2009

Mobile web resource tracking during a disaster or crisis situation

Phavanhna Douangboupha

Mobile Web Resource Tracking
During a Disaster or Crisis Situation

By

Phavanhna Douangboupha

Project submitted in partial fulfillment of the requirements for the
Degree of Master of Science in Information Technology

Rochester Institute of Technology

B. Thomas Golisano College
Of
Computing and Information Sciences

October, 1st 2009

Rochester Institute of Technology

B. Thomas Golisano College
of
Computing and Information Sciences

Master of Science in Information Technology

Project Approval Form

Student Name: Phavanhna Douangboupha

Project Title: Mobile Web Resource Tracking During a Disaster of Crisis Situation

Project Committee

Name	Signature	Date
Prof. Jeffrey Sonstein Chair		10 OCT 2009
Prof. Dianne Bills Committee Member		10/1/2009

Acknowledgement

I would like to express my heartfelt gratitude, appreciation, and thanks for one of the most important opportunities at the most critical time of my studies to the following people.

I would like to thank the following people for their patience, support, and kind assistance. Prof. Jeff Sonstein, Prof. Dianne P.Bills, Mr. Chris Denninger (RIT Public Safety), Prof. Tony Jefferson, Prof. Brian Ballsun-Stanton, Zack Panitzke, Mr. David Armanini (RIT Environmental Health and Safety), Ms. Lynn B. Daley (RIT Global Risk Management Services), Ms. Melinda J. Ward (RIT Global Risk Management Services), Mr. Chad Weeden, and Mr. Nicholas Paulus (RIT Publishing and Scholarship Support Services).

I would also like to thank the following people for their encouragement and their continued support. Prof. Edward Holden, Adwoa Boateng (College of Science Liaison/Librarian, RIT Libraries), Robert and Julie Norris, Corey and Chant Sent, Mike Galey, Andrew Kane, Abby Pacquing, Robert Lau, Kevin Brolly, Dave Morgan, Aliya Jaxiyeva, Mark Gabruk, my host family, and Regina Hovhannisyan.

I am thankful for Fulbright scholarship and it is such a great privilege to be awarded with the 2009 Southeast Asia Supplemental scholarship for this project. Thank you for giving me this opportunity.

Last but not least I would like to thank my family – my parents and my two sisters, Latthana Douangboupha, and Vannida Douangboupha.

List of Figures

List of Tables

1. Abstract

This paper proposes a prototype solution for a mobile web resource tracking system using mobile devices during an emergency situation. The system provides real time data to a decision maker so that he/she can effectively and efficiently monitor resources and assess the situation accordingly. Mobile devices (i.e., smart phones) support the ease of use for any location and at anytime. The Internet technology is selected to enable multiple or cross platform technology solutions for different mobile devices.

Resources in the scope of this project are human resources (e.g., a doctor, a police officer) and a chemical list in a room. With the use of a GPS-enabled device or a wireless-enabled device, the system is used to provide the current location of the human resources.

Transferring data between system databases and mobile devices is one of the important areas to address in this project. Since location data of a user is sensitive data, data should be protected via an encrypted protected network. In addition, because of the urgency of any crisis situation, it is critical that data from the system be able to be retrieved in a reasonable time frame. The investigation includes the exploration of database and data transfer solutions to meet the data availability during emergency conditions on mobile devices.

This document includes a description of the system design, a review of current technologies, proposed methodology, and the implementation. Review of the literature section provides background information on current available technologies that were studied (section 3). Four identified factors are suggested in the system design – usability, security, availability, and performance. The discussion of which technology is selected for each feature can be found in the implementation section 4. Proposed future research areas can be found in the conclusion section and recommendations for future work section.

2. Introduction

The idea of the project is to utilize mobile devices (i.e., smart phones), database, and Internet technologies during a disaster or crisis situation. The technology is a tool to monitor human resources and to provide a chemical data listing for an emergency team. The scope of the project includes research on factors related to the overall system design and implementation, the design system architecture for the system, and the creation of a prototype for demonstration. The human resource for the project is limited into four groups – doctor, nurse, firefighter, and police. The human resource tracking location covers an emergency site area; the RIT campus is used to demonstrate the concept. The resource assigned responsibility during the situation and contact information are selected as part of the scope. The assigned responsibility data is important to a decision maker to monitor the current status of how the situation is being responded to. The phone contact data are used to demonstrate how the system can be used to allow team communication using a smart-phone implemented function. RIT rooms are used as sample areas that can contain potentially harmful chemicals. The data is not the real data and is only used for the project demonstration. All data was inserted through a back-end (MySQL). The user interface for the system administration page is not part of the scope.

The prototype includes two main applications – location tracking and chemical listing module. Location- based data is one of the critical parts in the project to provide real time, current information to a decision maker, especially when urgency and timing are highly critical, replacing the need for manual personnel reporting. The chemical feature is designed to display potentially harmful chemicals in an emergency (e.g., fire) situation for the emergency team to look up.

The project aims to design an easy-to-use interface for end users to get important information as quickly as possible. One of the solutions is to have a pre-classification of emergency data. Only a pre-defined classified list of important data in the chemical listing, specifically during an off-line mode where there is no connection, is cached. Hence, only selected data is stored on the device.

The position tracking application consists of real time location update. As a doctor responds to the system alert, his/her GPS-enabled device will automatically send his/her current geographic X, Y coordinate location. The decision maker will have a virtual map showing real time position data of each team member on the site. Using Google map API technology, a virtual map is requested from a map server to plot each resource location on the map.

The two main target user groups are emergency response teams (human resources, for example, doctors) and decision makers (e.g., an emergency response manager, a director of public safety). The following are the two possible scenarios of use:

Scenario1, a fire fighter or a security officer has a need to know a chemical list stored in a room of a building. Instead of calling the central office and waiting for a response, the officer can browse through data of the room to see what dangerous chemical items are currently stored in the room. Consequently, he/she can decide if it is safe to enter the room and what precautions are required. The idea was formed during a meeting with Mr. Denninger (2009) of RIT Public Safety Department.

Scenario 2, a crisis manager on or off site browses information through the resource tracking system for the latest information about required human resources. Geographical positioning and wireless communication technologies are useful to the managers and users of the fleets on their status. It will support a specific subset of management decisions, for instance, how many medical

doctors have already arrived on site, how many more are required and how many more are on their way.

The system makes use of the Representational State Transfer (REST) web-service to bridge the connection between servers and clients (Richards, 2006). The REST web-service and the Internet solution for the prototype are selected to overcome mobile device cross-platform issues. The REST web-service provides a solution for the mobile device system to interface with different systems in manageable, lightweight implementation. REST utilizes Uniform Resource Identifier (URI) namespace technology that makes the application scalable, and the data can be easily shared by different servers.

3. Literature Review and Findings

There are technologies currently available that are useful to this project. Some identified core technologies include mapping, position tracking, information persistence, and resource management system. Following are some of the current solutions and technologies in the areas available to date:

- An intelligent and situation-aware pervasive system to support debris-flow disaster prediction and alerting in Taiwan (Kung, Ku, Wu, & Lin, 2008).
- A wireless first responder handheld device for rapid triage, patient assessment, and documentation during mass casualty incidents (Killeen, Chan, Buono, Griswold, & Lenert, 2006).
- Cost-based tracking of scheduled vehicle journeys (Tiesyte & Jensen, 2008).
- Sahana open-source disaster management system (Silva, 2006).
- Ushahidi web-based handheld device information distributing during a disaster by user generated content (Ushahidi, 2008).

- User finding location using mobile phones from Google called Google Latitude (Google Latitude, 2009) as of February 17, 2009.

An example of a resource management system for deployment during a disaster can be seen in the Sahana system. Sahana is an open-source disaster management system implemented with four main modules including a missing person registry, an organization registry, a camp registry, and a request management system (Silva, 2006). Other modules in the Sahana system consist of inventory management and catalog system, child protection module, volunteer management system, messaging module, and situation awareness. The Sahana organization registry module stores meta-data containing information about relief organizations involved in a particular disaster. The information includes types of services and support that an organization provides. By the same token, the Sahana request management system is available online to provide search solutions for customized aid catalogs, aid pledges, and requests. Another management module of Sahana is the volunteer management system. This module is used to record volunteer skill information, their availability, and to search for available volunteers based on skills. While the organization registry, the request management system, and the volunteer management system concern themselves with the management aspect of a disaster, the Sahana camps registry module uses Google maps to provide a Geographic Information Systems (GIS) view of an affected disaster area. Decision makers use the module to view location of camps.

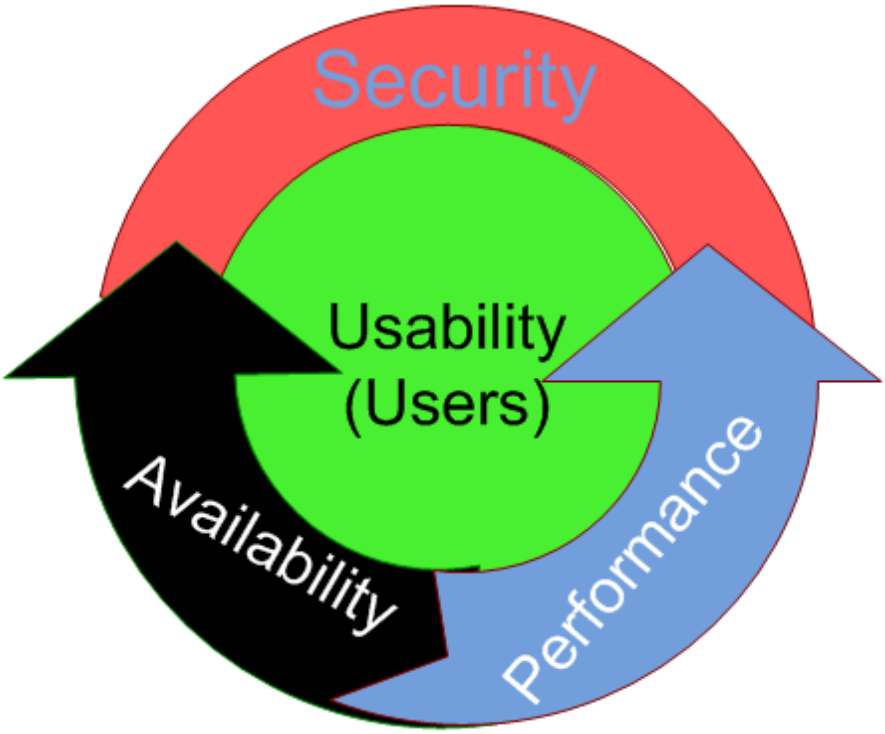
Apart from the small screen of a mobile handheld device, it also has access to limited wireless network bandwidth (Hu, 2009). Network availability and connectivity (e.g., Wi-Fi and 3G network) could be an issue for mobile devices especially during an emergency situation. The network traffic is very likely to be higher than normal, or there could possibly be no network connection available at all. Hu (2009) identifies that for a mobile system, an authorized user

should have reliable and timely access to content. Therefore, having a fall-back version in which data is stored temporarily or permanently on the device itself will allow the application to run even after the connection is no longer available. Nevertheless, this will increase system vulnerability to physical attack if the device is stolen.

One of the reasons for selecting mobile platform technology for the system is to allow data access during an emergency situation anywhere. A selected solution should have a reasonable system performance. Consequently, design and implementation for a reasonably fast system is also discussed in this paper. Users are often on the move while viewing content on their mobile devices. Therefore, Few (2006) and Golding (2008) identify that the presentation layer or the user interface design on a small screen of a smart phone is a deciding factor in a mobile web application. Unlike the normal web desktop screen size where people generally do not look at all of the content on a page, the mobile device screen is small enough that people can view the entire content on the screen. Hu (2009) states that the Internet browser on a mobile device (i.e., Microbrowser) has fewer functions compared to its desktop browser counterpart. On the small screen, users only quickly glance at the screen for content which they specifically want (Krug, 2005). In an emergency situation, it is critical for information to be presented clearly on a mobile device. The intended users should not have to spend a lot of time and effort to learn to use the application. Sauter (2009) believes that usability of a mobile application should not suffer from the choice of security and privacy consideration. Cranor and Garfinkel (2005) believe that security is very important; however, it is a supporting task where the primary goal for a software application is usability. Warsi (2007) describes the usability and performance of a web application and of a mobile phone application. Usability is, therefore, believed to be the centre of system design and implementation for this project.

There are many factors that must be considered in designing a web site displayed on a mobile device. From the above discussion, the four main factors that are vital to the system are identified as follows: usability, security, performance, and availability (as Figure 1 suggests).

Figure 1: Four main design factors in the system design



Martin (2009) and Raton (2004) address the issue of security for a wireless application. Ballad and Ballad (2009) point out the security concerns for the World Wide Web technology. Since the system is intended to be used during an emergency situation, there is a concern about unauthorized interception from hackers. Therefore, security is of vital importance to the system. It is represented as the top red area in Figure 1. On the other hand, availability and performance are selected as the important secondary issues. The time it takes the system to run the security computation code in a small device with a limited memory space and the constraint of network connection can be critical (Kambourakis, 2008). However, the higher the security is, the longer it will take for data to be retrieved and transferred (i.e., less performance). Similarly, if data is

available openly to the public, there is more risk to attack (i.e., less security). The more sensitive the data being stored on a device, the less secure it is, especially in the case of the device being lost or stolen. As discussed above, usability is the centre of the design and is the most important factor to be considered. The security, availability, performance, and usability solutions selected for this project and their trade-offs are discussed in the implementation section.

3.1. Usability and User Interface (UI) Design on a Mobile Device

Society is evolving at an ever-quickening pace with large volumes of data available to users. One method of tapping into this data is by using mobile devices. Mobile devices allow the user to get access to key data from any location and still receive continual updates of this data. This allows the user to move quickly as a situation evolves. This section discusses human factors and suggests a user interface design (UI) that aggregates existing techniques such as the information dashboard. According to Few (2006), an information dashboard is a way to display information on a single screen display. The purpose of the information dashboard is to efficiently monitor the information needed by a user so that he/she can achieve his/her objective (Manson, 2007). The design utilizes a visual display of the most important and relevant information that can be fit entirely on a single screen display so that a user can view the information at a glance. Simplicity is said to be the key to a successful UI design.

The design for a mobile web application should be implemented in a way that can be used to get people's attention at first glance. In addition, the design needs to address the way humans interact with information on a small screen. It can be difficult to predict the condition or environment of a user when he/she uses a mobile device. The factors include user environment, user knowledge, language, type of content, context of use, input devices, and available technologies. There have been many studies that address various factors in the usability (Ham et

al., 2008). Still, these factors can change over time and it is difficult to predict trends. Nevertheless, human factors and physiological limitations are the most standard factors that can be used as a foundation in measuring mobile usability.

Karlson et al., (2008) state that the trends of mobile device usage are moving in a direction where richer content is viewed through small input and output channels. Therefore, to meet the needs of mobile device users, the studies that focus on how people perceive information during a short period of time on a small screen should be examined.

Based on the discussion above, following are the decided four factors considered in the usability and UI design on a mobile device for this project:

- Use of color for usability (Few, 2006)
- Human factor on memory (Few, 2006; Hoekman, 2007; Penn State, 2008; & Remaid. 2005)
- Human factor on eyes movement (Lehikoinen et al., 2007; Ruel & Outing, 2004)
- Human factor on familiar knowledge (Few, 2006; Krug, 2005; & Lehikoinen et al., 2007)

3.1.1. Use of Colors for Usability

Few (2006) suggests the use of alternate colors between items on alternate lines in an information dashboard screen. The technique is used in the chemical listing table room viewing screen to make each row of the table presented as individual values as can be seen in Figure 2. The emphasis colors are used minimally and only when they are needed so that it is easy for human eyes to look at and the important information is distinctively shown. Similarly, for users to be able to skim through the information efficiently, the design should be a simple one (Lehikoinen et al., 2007). The use of space between lines of text and items is also recommended to relax eyes and increase user attention to the content (Öquist, 2008).

Figure 2: Make use of alternate fill colors to delineate rows in a table

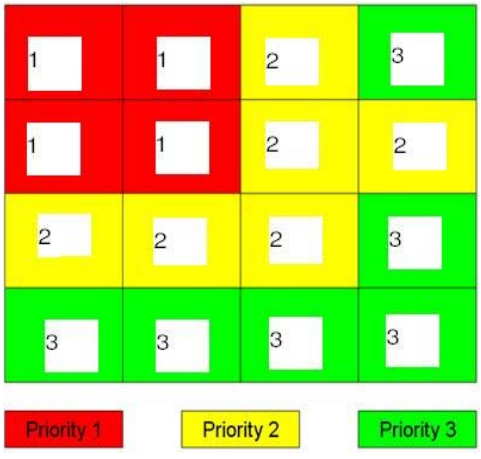
3.1.2. Human Factor on Memory for Usability

According to a PennState study on human factors and HCI (PennState, 2008), there are four different types of human memories including iconic, short term, working, and long term. Both the PennState study (PennState, 2008) and Few (2006) agree that short term memory is the primary working memory. It stores temporary information, and the information resides there during conscious processing. This is the memory that people use the most when they use their mobile devices to quickly look for information. Few (2006) states that people can store from three to nine items in the short-term memory. Similarly, the PennState study (PennState, 2008) indicates that the limit on peoples' short-term memory is five items, plus or minus two. Therefore, it is suggested that the information a user is looking for should be reached within three to five links, where each link should be interlinked. Interlinked means each link to a new screen or a new page should have a connection or keyword displayed in the very first priority viewing area (on the top left corner of the screen). Since humans can remember up to seven items at a time (similar to how they remember a phone number), it is suggested that the number of control icons or menus should be limited to about seven items. Hoekman (2007) concurs on the idea of using information dashboards for UI, and he suggests including only those features

that are absolutely necessary to users. It is said that people tend to only learn about 20 percent of what an application is capable of doing and if they can they would ignore or hide about 80 percent of the features. Some users would just ignore them totally (Hoekman, 2007 & Krug, 2005).

Figure 3 shows the priority areas of human page scan on a normal website desktop screen display. Priority number one is the most permanent area of content that users see. Subsequently, people tend to view the areas of priority number two and three respectively. Despite the fact that the screen of a mobile device is smaller than a traditional desktop screen size, people still have the same viewing pattern (Lehikoinen et al., 2007). Therefore, the information should also be arranged according to its priority on the mobile screen. Instead of leaving a page, having a pop-up window on top of an originial page is selected as another way to flash information to users many times continuously in this project (Figure 2). The pop-up window that is displayed on top of an existing screen allow users to see some of the original information such as a heading. This would help users to make a connection on their current task of the pop-up window from its orignal screen.

Figure 3: Priority areas of human eye scan toward web site content[1]

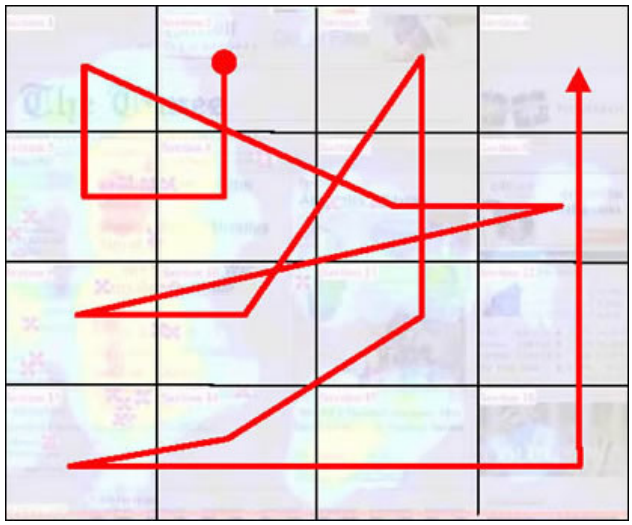


[1] Taken from Ruel, L., & Outing, S. (2004, September 8). Viewing Patterns for Homepages. In Eyetrack III: Online News Consumer Behavior in the age of Multimedia [a project of the Poynter Institute]. Retrieved May 7, 2009, from http://www.poynterextra.org/eyetrack2004/index.htm

3.1.3. Human Factor on Eyes Movement

Lehikoinen et al., (2007) states that a good layout design is the layout that is easy for human eyes to scan through. To accomplish such a task, one has to understand the way human beings view screen information. It is currently understood that people's eyes move in a Z-shaped pattern motion (as can be seen in Figure 4 when they skim a page for reading. Hence, any important information such as icons or controls for the next task should be placed in this viewing path, while the remaining "lower priority" data should draw the interest of the viewer through other means such as color or weight of text. Displaying a visual feedback cue that helps users to recognize what choice is currently being selected is a way to help users navigate to website content (Lehikoinen, 2007).

Figure 4: Typical Z-shaped pattern motion human eye movement when looking at a desktop web screen[2]



[2] Taken from Ruel, L., & Outing, S. (2004, September 8). Viewing Patterns for Homepages. In Eyetrack III: Online News Consumer Behavior in the age of Multimedia [a project of the poynter institute]. Retrieved May 7, 2009, from http://www.poynterextra.org/eyetrack2004/index.htm

3.1.4. Human Factor on Familiar Knowledge for Usability

Few (2006), Krug (2005), and Lehikoinen et al., (2007) agree that users can quickly learn to use a new application or can quickly scan through a page when they are already familiar with the design or have been used to that standard. This makes users feel productive and smart (Hoekman, 2007 & Krug, 2005). Lehikoinen (2007) and Hoekman (2007) point out that a user's previous skill and knowledge should be fully leveraged since this promotes familiarity and consistency. Hoekman (2007) further suggests that people tend to learn one way of doing something and are more likely to stick to the routine. In addition, a user interface designer should also include support for customization and personalization features that meets a user's own style of working. Hence, it is recommended that targeted users should be identified prior to developing the UI. This technique can be employed to identify unique requirements for each targeted device based on the user population and to develop different styles or adaptive styling according to each device with which the user is already familiar with the design. This will support smooth transactions from the device interface to the web page interface. Consquently, users will have fewer new skills to learn in order to use the interface. For example, the iPhone screen design has a square button for each application and makes use of a black background. This format can be applied to other mobile device user interfaces, for example, Google Android smart phone. Android mobile web interface can be utilized with a white background and has a hidden menu for control to imitate the Android default view. This could provide a good starting point for Android users since they have already adapted to how the UI looks and what to expect.

3.1.5. Field Study and User Testing for Usability Using Information Dashboard

To evaluatate the design approarch selected for this project, thirty-one (31) people were observed testing a mobile web application prototype for this project as evidenced during the Imagine RIT festival in May, 2009. The main features of the prototype include:

- A weekly and daily weather update;
- a building floor plan and room list that contains a list of chemicals in each room;
- the chemical list is linked to a Material Safety Data Sheets (MSDS) database;
- human resource locations on a Google map with the viewing options of marker pop-up viewing profile and contact information of a user. The screen shot of the location tracking feature is shown in Figure 5.

Figure 5: The initial prototype of a mobile web application (map page) on a resource tracking during an emergency situation using iPhone and iPod Touch user interface design

The application is a web application that works on normal desktop, iPhone, and iPod Touch devices with Internet access. Seventeen (17) of the subjects completed a survey after the observation. These are users who are interested in the application and are familiar with the World Wide Web (WWW) application. Three users already had an iPhone or iPod Touch, and two of the seventeen users had an interest in buying an iPhone. Four of the additional observed subjects were targeted, and they were part of the RIT emergency response team. The rest of the general

users tested the prototype after a short demonstration of how to use the system. These are general public users who did not have an interest in using the application.

Church et al., (2007) point out that the nature of mobile content is much shorter in length than the WWW content. This introduces challenges in presenting content to users in a text format, whereas a graphical presentation (when possible) could provide much more meaningful data to users. Consequently, this satisfies the requirements of continuous mobile use where mobile users use a device to quickly view information. Content is presented at a glance in a single screen to allow users to quickly skim through the data and locate the information that meets their needs (Krug, 2005). Unfortunately, there is a size constraint for mobile device displays. Therefore, the UI has to utilize as much of the limited area as possible and be able to apply a suitable graphical representation of data.

To design a UI for an application, there are many factors that must be taken into consideration regarding the target users. These include user environment, language, infrastructure, data access speed, and culture. Fortunately, the visualization standard from the information dashboard can at least be used to overcome language barriers by visually displaying information. Few (2006) suggests that the data on the screen should be selective. The pixels of the most important data should be enhanced, whereas non-data pixels should be reduced. Consequently, if the practice is successfully applied, the speed of loading information could also be improved by using emphasis colors for the most important information and by removing unnecessary pixels.

The result of user testing on the iPhone web resource tracking supports the theory of human factors on memory during Imagine RIT. None of the users were interested in the daily and the weekly weather update feature of the application. On the other hand, they were very engaged in

chemical viewing information of a floor in a building, the human resource profile information, and the feature on making a call directly to an emergency team member.

3.2. Security Factor

The system consists of three main components (i.e. three-tier) including the web server, the database server, and the client device. The network connection between the server and the client is another vital part for a mobile application. In addition to the web security, using mobile devices also introduces wireless networking security concerns. The wireless environment is less secure than a wired environment, and there are more possibilities for attacks (Martin, 2009 & Raton, 2004). Three (3) primary security factors are identified to be a vital part in this mobile web emergency application:

- Data transmission security (Martin, 2009; Raton, 2004; & Pashtan, 2005)
- User integrity (Ballad & Ballad, 2009; & Erl, 2009)
- Data security (Pernul, 2008 & Priebe, 2008)

3.2.1. Data Transmission Security and User Integrity

The two key issues with wireless network security include the over-the-air transmission and the additional gateways between wireless and wired domains (Pashtan, 2005). Data that is being transmitted between different intermediaries can be intercepted and viewed by unintended recipients. Therefore, sensitive information should be encrypted to prevent any unauthorized activities (Erl, 2009 & O'Neill et al., 2003). There are many encryption algorithms; some of the encryption techniques recommended by the experts are:

- Digital signatures – Hash-Based Message Authentication Code (HMAC) and Secure Hash Algorithm (SHA-1) (Erl, 2009 & Richards, 2006);
- Public key cryptography – Advanced Encryption Standard (AES) and Rivest Shamir Adleman (RSA) security (Pritchard, 2003);

- Extensible Markup Language (XML) encryption (Richards, 2006);
- Network or Firewall security – Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols (Kambourakis, 2008);
- Password-based system protection (Ballad & Ballad, 2009);

Dawson, Winterbottom, and Thomson (2007) suggest that data encryption and data integrity checks are necessary to protect privacy of users in such a system. The data integrity check is used to protect the data from being tampered with by unauthorized users. By the same token, the data encryption technique also ensures that only authorized users can view the data. To verify the origin of data transmitted from a server, Erl (2009) and Richards (2006) both recommend the use of digital signature techniques to confirm the origin of the data (e.g., HMAC - Hash Message Authentication Code). HMAC algorithm can be used to encrypt data before being sent over a network. The encrypted value is decrypted with a client secret key. With HMAC, both the client and the server must have a secret key. The XML format for the encrypted data can be seen in Figure 6 and its equivalent XML location data can be seen in Figure 7. Richards (2006) shows a generic HMAC algorithm as can be seen in Figure 8.

Figure 6: Encrypted XML current location data

```
<Name id=”xxx123”>Phavanhna User</Name>
<EncryptedData
Type=’http://www.w3.org/2001/04/xmlenc#Element’
xmlns=’http://www.w3.org/2001/04/xmlenc#'>
<CipherData>
<CipherValue>X11X22X33</CipherValue>
<CipherReferenceURI>http://
xxx/location/geocoding/cipher</CipherReference URI>
</CipherData>
</EncryptedData>
</LocationInfo>
```

Figure 7: Equivalent XML current location data as plaintext

```
<!—ID represents the device unique
identification!-->
<?xml version='1.0'?>
<LocationoInfo
xmlns='http://location/geocoding/current'>
<Name id=”xxx123”>Phavanhna User</Name>
<current>
<longitude>-104.441056</longitude>
<latitude>32.176991666</latitude>
<Point>
<coordinates>-
104.441056,32.176991666,0</coordinates>
</Point>
</current></LocationInfo>
```

Figure 8: A generic HMAC algorithm taken from Richards (2006), Chapter 12: XML Security, Introducing XML Signatures

```
unction hmac ($key, $data)
{
$b = 64; // byte length
if (strlen($key) > $b) {
$key = pack("H*",sha1($key));
}
$key = str_pad($key, $b, chr(0x00));
$ipad = str_pad('', $b, chr(0x36));
$opad = str_pad('', $b, chr(0x5c));
$k_ipad = $key ^ $ipad ;
$k_opad = $key ^ $opad;
return sha1($k_opad . pack("H*",sha1($k_ipad .
$data)));}
```

HMAC utilizes SHA-1 algorithm with the addition of key encryption techniques. According to Kambourakis (2008), a selected data in the Wiki mobile web system is encrypted in XML format and secure sessions. Wiki uses both asymmetric and symmetric cryptography. Wiki also utilizes two encryption algorithms - advanced encryption standard (AES) with key length of 256 bits and Rivest Shamir Adleman (RSA) with key length of 1,024 bits. The AES key is used for symmetric encryption, whereas the RSA is used for public key operation. According to Pritchard (2003), RSA is not bidirectional, and it is not applicable if the client has no way to generate a private key. Prichard (2003) suggests HMAC or Message-Digest algorithm 5 (MD5), since it is a faster method for encryption. Wiki clients receive the server's public key via email in the form of a plain text base 64 encoded X.509 certificate issued by a Certificate Authority (CA). Despite the fact that HMAC is utilized in the mobile web application such as Wiki, HMAC encryption operation is computationally intensive; it can affect the system resource usage (Erl, 2009). Therefore, for the mobile power consumption issue, only selective or partial data should be encrypted for location XML file in the future (Kambourakis, 2008). The map page for this project already consumes a large loading file – Google Map API; HMAC is therefore not applied to the location file. The sensitive chemical data for off-line mode is encrypted through SSL protocol of RIT Gibson server. Consequently, HMAC is not used in this project. While HMAC is the data encryption technique recommended by Richards (2006) and Erl (2009) for data origin authentication, Ballad and Ballad (2009) believe that SHA-1 (one-way encryption) is suitable for PHP web applications in the user authentication feature. One-way encryption is the encryption technique where no decryption is required. This technique is, therefore, applied in the user authentication module for this project where the user password is encrypted both at user input and the actual user password is stored in the database itself. The data is checked and validated in the encrypted format.

Andrew and Whittaker (2006) state that a session key can be hijacked by an attacker. The attacker can scan for a valid session key with a shell access or the cross-site scripting attack to a target web application. Erl (2009) indicates that session-based encryption with a short life span allows protection against an attack that might intercept messages. Andrew and Whittaker (2006), Ballad and Ballad (2009), and Kambourakis (2008) agree on the random generated number for session key or token key. This ensures responding and checking legitimate authentication of login. The random one-time key is therefore generated every time an authorized user is logged in to the system. The key is created and stored in the database. The recorded key for each user can be used to check if the client request is from the authorized user not a hijack request from somewhere else. More detail on how the key is used can be found in the implementation section (4.2.1).

Ballad and Ballad (2009) recommend the use of a CAPTCHA (Completely Automated Turing Test To Tell Computers and Humans Apart) to differentiate between the legitimate human users and machine- created accounts. The technique is a solution to protect the system authentication from brute-forcing passwords. However, Andrew and Whittaker (2006) point out that there is already software called Optical Character Recognition (OCR) to guess a CAPTCHA and pretends to be a human user. More importantly, the recognition technique could make it harder for users in an emergency situation where timing is very critical. As a result, this technique is not utilized for the project.

In addition to the encryption technique, Kambourakis (2008) proposed two other security technologies for Wiki that enable legitimate mobile web users to access the system - SSL and TLS technologies, that can be used to ensure confidentiality in their three-tier system and Peer to

Peer or P2P system to disseminate data on different machines in the Wiki system. SSL can be configured through the web server such as Apache or through a Certificate Authority (CA) such as VeriSign. Erl (2009) also recommends using SSL and TLS technology to protect data at the transport layer. SSL is normally used for sensitive data such as a credit card numbers (Carroll & Broadhead, 2001), (Ferraro, 1998). SSL comes with different levels of encryption; the more bits of encryption, the harder it will be to hack into (VeriSign, 2005). After registering with a provider, SSL will be enforced after the private and public keys are verified as a matching pair (VeriSign, 2005). SSL verification takes place between client browser and server where the browser shares a secret key. The server examines the browser's digital certificate (X.509) for authentication. Similarly, the browser also authenticates the server by checking the server's digital certificate. For the future of encryption, we will see more and more key management systems being used with encryption such as Digital Rights Management Systems (DRMS) (Nützel & Beyer, 2006) and more of the technologies are being adapted to mobile devices. In this project, the chemical off-line module is being pulled from RIT Gibson web server where it is protected via SSL.

Utilizing code encoding techniques for an application message is another technique used to protect client-side code. Johnson et al., (2007) suggest that replacing the application messages with part-Unicode or hex strings can help to protect the JavaScript code since the code will become less human readable due to the encoding.

To protect the system against cross-site scripting attack (XSS), NULL string attack, buffer overflows and canonicalization, Ballad and Ballad (2009), and Andrew and Whittaker (2006) suggest checking all possible script code that could potentially be entered in an input form.

Checking any input string from the user in a form before it transfers over to the server can be used to filter out a bad input. Validating all possible supplied checking strings that can be used to capture all possible characters of a certain hack string (Ballad & Ballad, 2009). For instance, NULL string attack should not only be checked against the empty string but also other representations including "\0," "%00," and "\\00." The character validation checking technique is selected to validate user input for the login authentication module (Andrew & Whittaker, 2006).

3.2.2. Data Security

Hu (2009) addresses the concern of handheld data protection if a mobile device is missing or stolen. Having a policy that forces users to set a password user identification device protection is suggested by both Hu (2009) and Geddis, Adler, and Brouwer (2009). iPhone third generation has the option of data encryption for data backup on the device (Geddis, Adler, & Brouwer, 2009). Another action that can be taken to protect the data on a mobile device is to remotely delete data if it is authorized by the rightful owner of the device. However, the user must register the device with a service provider. Hu (2009) presented a remote lock down device as one of the solutions being used in the handheld security by Hewlett-Packard Development Company and Sybase Inc (the embedded database vendor). According to Hu, a registered owner of a device can call the service provider to lock down his/her device remotely. Similarly, Apple offers the functionality to remotely delete data from a stolen iPhone device for 3GS version (Geddis, Adler, & Brouwer, 2009). Frakes (2009) looked at how iPhone 3GS data can be remotely wiped clean. However, to perform such a feature, users must have a MobileMe email account through iPhone/iPod Touch's Mail application. In addition, the operation is only possible while the device is currently connected to the Internet. The remote data protection technology is not utilized in this project. Nevertheless, the technology can be further investigated.

In terms of security for a relational database system, Priebe (2008) and Pernul (2008) suggest the use of role-based access control (RBAC). This system architecture is fundamentally a web system, however; the web system is stateless. Therefore, any data that is required to be accessed from multiple pages will have to be stored locally or temporarily. Andrew and Whittaker (2006) suggest that any important computing or validation on the client should be confirmed by checking its match on the server before allowing access to the system. This includes the input data passed from the client. In addition, the system should only provide data that a user needs or only to what he/she is allowed to see - least privilege technique (Chen, Dunagan, Verbowski, & Wang, 2005). Likewise, Geddis, Adler, and Brouwer (2009) suggest keeping sensitive data on the server and only storing non-sensitive data locally on a mobile device. While the remotely lock down device, remotely delete data, and data encryption were identified as a good handheld security technique as discussed previously, Zdziarski (2008) demonstrated that some very important information and data can be recovered from iPhones including browser history and caches, even when the caches were cleared and deleted (Vance, 2009). Despite the fact that the remote data protection technique can be utilized in the future, still for a high security application extra precaution should be investigated. As a result, there are two different options to view the chemical listing in this application to address the security issue. One is the list that can only be viewed while there is a network connection - data is being pulled directly from the server transferred as XML file (please see the implementation section for more information). The other option is to view and to cache the data on the device to be viewed later in the case of need when there is no connectivity.

3.2.3. Other Security Solutions

If it is required, security training should be provided to staffs as verified by the BS 7799/ISO 17799 information security management standard as a best practices to information security (Bisson & Saint-Germain, 2007). IBM uses a technique called "information agenda," which includes delivering information on demand. IBM suggests that everything has to be agile; this means a security solution should be adapted with technology and to meet the needs of users (IBM, 2009). These solutions are not implemented for this project as it is a prototype application; nevertheless, security solutions could vary according to the security policy of the organization that might adapt the application. Therefore, it is recommended to select the security solutions that meet the needs of target users.

3.3. Optimization

Golding (2008) and Lehikoinen et al., (2007) suggest that users are less tolerant of delay in a mobile-services context. Consequently, the mobile web performance is another important issue to be considered in addition to usability and security. Mehta (2008) shows the minimum expected feature for mobile web development by The World Wide Web Consortium (W3C) as can be seen in Table 1.

Table 1: W3C-Defined Default Delivery Context (DDC)[3]

Context	Size
Usable screen width	120 pixels minimum
Markup language support	XHTML basic 1.1 delivered with content type application/xhtml+xml
Character encoding	UTF-8
Maximum total page weight	20 kilobytes
Colors	256 colors minimum
Style sheet support	CSS level 1; CSS level 2 @media rule together with the handheld and all media types

Following are the list of technologies that are suggested for optimization by King (2008) and Johnson, White, and Charland (2007):

- CSS optimization: Utilizing CSS technology to replace JavaScript behavior; using image merging or CSS sprites to load images (Johnson, White, & Charland, 2007; King, 2008)
- Ajax optimization (Johnson, White, & Charland, 2007; King, 2008)
- Image optimization (Thus, 2008)

[3] Taken from Mehta, N (2008), p 20

- Using compression – HTTP, JavaScript, Gzip, and YUICompressor (King, 2008 & Thus, 2008)

According to King (2008) and Johnson et al., (2007), CSS optimization will help to make a web site load faster. Some CSS techniques were used to replace JavaScript behavior as recommended by King (2008). King (2008) explains that substituting JavaScript with CSS is a way to reduce a web page overhead caused by HTTP requests. A CSS technique on hover is used to replace JavaScript behavior effects on rollover, and instead of using table in HTML, lists are used. These techniques are utilized specifically for the view and save mode page.

CSS sprite technique is a well-known technique for a web optimization. CSS sprite is also known as image merging (Johnson et al., 2007). CSS sprite image technique is used in the map page for location tracking map feature. King (2008) and Johnson et al., (2007) suggest the CSS sprite optimization for a web page to reduce HTTP requests. The technique reduces loading time by grouping multiple images into one file. The technique is used by Youtube, Yahoo.com, and AOL.com. The file contains composite images and hence it only needs to be loaded once instead of having multiple files. Each image is displayed as a CSS background and is selected via pixel positioning of the composite file (King, 2008).

Group selectors with common declaration technique use common CSS properties in a grouping manner. It is applied for CSS properties for CSS files in this project (King, 2008). This technique reduces the number of CSS lines of code.

Ajax optimization is also recommended by both King (2008) and Johnson et al., (2007). King (2008) states that unoptimized Ajax functions can harm the system by causing performance lags. Some suggestions from King (2008) and Johnson et al., (2007) for JavaScript and Ajax optimization are:

- Remove JavaScript comments

- Reduce whitespace
- Use JavaScript Shorthand (e.g., x++ instead of x = x+1)
- Use string constant macros
- Shorten user-defined variables and function names
- Remap built-in objects (e.g., var w = window; and use "w" variable throughout the code instead of the full variable name "window")
- Apply inline localized functions, assume default values

  e.g., xhr.open("GET","Authentication.php") instead of

  xhr.open("GET","Authentication.php",*true*))
- Combine all JavaScript files into one file, use lazy-load to pre-load JavaScript and CSS files on to a client's machine for future use
- Minimizing HTTP request

Ajax was used in the project to improve the system usability via user interaction. In addition to the usability point of view, Ajax is also utilized at the front-end to protect the system via validating user input in the login form before even triggering or requesting the back-end check using PHP code on the web server.

Ajax optimization techniques are selectively used for the off-line module where security and file size are critical. On the other hand, Ajax and JavaScript code were implemented following the code-guiding standard best practices with minimum optimization to maintain code readability for future developments for the map module and the chemical view online module.

Optimizing images is used to improve a web page loading performance as suggested by (Thus, 2008). SiteReportCard (http://sitereportcard.com/imagereducer.php) and DynamicDrive

(http://tools.dynamicdrive.com/imageoptimizer/index.php) were the two free online image optimizer sites used to reduce the size of image icons for the map application.

King (2008), Saravana (2009), and Thus (2008) suggest web compression for web page optimization. Gzip is another way to compress a file and hence reduce the HTTP response file size (Azad, 2007). However, the Google live map related content is not compressed because it is loaded externally from Google API. The Gzip configuration in Apache using htacess file can be seen in Figure 9. Figure 23 and Figure 24 show the map and the chemical listing page content. The compressed content is also shown.

Figure 9: Gzip output compression in Apache using htaccess file

```
<IfModule mod_gzip.c>
mod_gzip_on       Yes
mod_gzip_dechunk  Yes
mod_gzip_item_include file      \.(html?|txt|css|js|php)$
mod_gzip_item_include mime      ^text/.*
mod_gzip_item_include mime      ^application/x-javascript.*
mod_gzip_item_exclude mime      ^image/.*
mod_gzip_item_exclude rspheader ^Content-Encoding:.*gzip.*
</IfModule>
AddType text/cache-manifest .manifest
```

3.4. Location Tracking

The proposed prototype system depends on a location-based technique to determine a real time position. In general, there are four geo-location methods including triangulate, associate, geo term extraction, and data entry or geo-coding.

Mallick (2003) states that there are several available mobile positioning techniques that include network-based solutions and handset-based solutions. Katsaros, Nanopoulos, and Manolopoulos (2005) suggest the satellite-based system, also known as GPS and the Terresrial-Infrastructure-based system, also known as the network centric approach. GPS mobile positioning is applied by both tracking of scheduled vehicle journeys by Tiesyte and Jensen (2008) and the intelligent and situation-aware pervasive system in Taiwan by Kung et al., (2008).

According to Bellavista and Corradi (2007), Katsaros et al., (2005), and Mallick (2003), the geo-coding position technology relies on getting the location of an object via street address or zip code and converts it into a meaningful X, Y coordinate or a latitude and longitude coordinate format. In fact, any available mapping tool such as Microsoft live search maps, Google maps, and Yahoo maps require the geo-coding coordinate system to plot a location on the map.

The Global Positioning System (GPS), cell towers, and Wi-Fi positioning service (WPS) are the three well-known techniques to identify a mobile device geo-location (Mark & LaMarche, 2009). GPS and cell towers are based on triangulation to identify an object position by using the locations of known objects (B'Far, 2005).

The proposed system can request mapping information from an existing map server such as Google maps or Microsoft live search maps. Hillborg (2002), and Purvis, Turner, and Sambells (2006) state that geo-coding services respond to a request in a form of the Extensible Markup

Language (XML). Google maps (Google, 2008) support the file format in XML, and Microsoft live search maps (MSDN Blogs, 2007) has application to support direct file importing for a map information request using the Keyhole Markup Language (KML) format. Supplying coordinate information of an object in XML format to the map servers enable data to be processed faster than if it is requested directly.

Olla (2008) explains that GPS relies on satellites that send information (microwave signal) back to the earth. The information is a navigation message of each satellite's position and time. An object location is calculated by GPS receivers by the use of triangulation, and the signal information is provided by at least three satellites in order to determine an object's location and four satellites for greater accuracy (Olla, 2008). The use of the fourth satellite enhances the accuracy to the order of nanoseconds. GPS receivers compare the time difference between the arrival of satellite signals to tell the position (Katsaros et al., 2005; Olla, 2008). Olla (2008) explains that three satellites are needed in the calculation because the first satellite provides possible locations of an object narrowed down to the surface of a sphere. The position is recorded as a radius equal to Range 1. By the same token, satellite number two provides confirmation that the object is located within the first sphere (as located by the first satellite). Satellite number two provides additional position circle of a radius Range 2. The first and second satellites indicate that the object is positioned between the intersection between their two spheres - sphere one and sphere two. Finally, the third satellite shows a third sphere of radius Range 3 for the positions. The object position is the intersect location of the three spheres. The fourth satellite is used to confirm the location and hence provides the time reference. According to B'Far (2005), a GPS-enabled device provides geo-code location accuracy at about 1 to 5 meters. Despite the fact that the GPS system is widely used with many position tracking systems, it does come with some drawbacks. GPS does not work within indoor environment positioning, and it requires

costly power-consumption on mobile devices (Katsaros, Nanopoulos, et al., 2005). In addition, not all mobile devices are GPS-enabled.

Similar to the GPS system, cell towers location technique requires a known distance between three different cell towers within range of the device (Olla, 2008; Wu & Tseng, 2007). Otherwise, at least two cells towers are required. Each cell tower with a unique cell identifier returns the device request with its position data. Each mobile phone constantly pings a signal to nearby cell towers to get the cellular radio signal. Having the three cell positions' data, an algorithm can be used to calculate the final location of the mobile phone which lies in the middle of the three points (B'Far, 2005). The advantage of the cell tower triangulation technique is that it is available for all mobile phones, and cell towers can be set up to send signals to mobile devices. Unlike GPS, cell towers will work in both indoor and outdoor environments. The accuracy of the position allocation depends on the density of cell towers in the area (Mark & LaMarche, 2009). To request a geo-location from a mobile phone service provider (or a cell tower), four required parameters are the cell tower identification number, the four digit Location Area Code (LAC), two to three digits of the Mobile Network Code (MNC), and three digits of the Mobile Country Code (MCC) (Naresh, Pingali, Varma, Krishna, & Venkata, 2008) as can be seen in Figure 10.

Figure 10: An example to access location information from native applications of Nokia (Laineste, 2009)

```
Nokia phones

MCC:System.getProperty("com.nokia.mid.countrycode")
MNC:System.getProperty("com.nokia.mid.networkid")
Cell id: System.getProperty("com.nokia.mid.cellid")
LAC(Location area code): System.getProperty("com.nokia.mid.lac")
```

Wi-Fi Positioning Service (WPS) is the least accurate technique for location finding among the three techniques (GPS, cell towers, and WPS), but the data can be retrieved quickly from a

server. An IP address from a mobile device Wi-Fi connection is used to get a "guessed" location back from a service provider database such as Google map (Mark and LaMarche, 2009). Nevertheless, Google mobile map and Navizon peer-to-peer wireless positioning are able to use their massive data collection to provide a better location result for a device. A non-GPS-enabled device can also access the services to locate their geo-coding location.

Some of the available and well-known mobile device location locator technologies are Google Gear (mobile device Google map), iPhone Core Location, and Navizon. They use all three triangulation technologies to get the most accurate result. Google gear and iPhone Core location are designed to provide the best accurate results according to a user request (Google, 2009; Mark & LaMarche, 2009). Core location is specifically programmed for iPhone. Google gear is compatible with many mobile devices including Windows Mobile and Android, but not all mobile devices (e.g., iPhone) support Google gear. Navizon is free for a cellular enabled-device, with some limitations using cell ID positioning, but it is not free for a Wi-Fi or cellular mobile phones (Mexens, 2005-2008). Navizon database collects geo-coding data from those users who have GPS-enabled devices. An alternative technology is a development tool called PhoneGap that utilizes web application technology and Objective-C core features available on three mobile devices - iPhone, Android, and Blackberry (Berka, 2008).

3.5. Database Investigation

Currently the SQLite database is supported by Safari 3.1+ and iPhone OS 2.0+ or WebKit (http://developer.apple.com/safari/library/documentation/iPhone/Conceptual/SafariJSDatabaseGuide/UsingtheJavascriptDatabase/UsingtheJavascriptDatabase.html#//apple_ref/doc/uid/TP40007256-CH3-SW1, 2009). SQLite has the advantages of being open source, lightweight, and having portability. In the iPhone mobile Safari, SQLite is used for file caching. Newman (2005) also suggests that SQLite is ported to Palm OS. SQLite is used as a secondary database to store data on a client or a device. In addition to its lightweight property, SQLite allows fast internal data manipulation via SQL commands (Newman, 2005). SQLite is suitable for small searchable data, because it provides fast, random access to the data instead of reading the whole file into memory before an element can be read, like XML database format (Newman, 2005). However, SQLite database does come with some limitations – file size or scalability, security, file locking issues, and current support.

First, the size of SQLite database is limited to 2GB in a single file (Newman, 2005) and each operating system also has different size limitations. For instance, in an iPhone or iPod Touch, SQLite maximum size is limited to only 6 MB (Apple Inc., 2009). Consequently, only selected data should be pulled from the main database to store on SQLite database. Second, SQLite is a HTML5 file system database where the security is based on the file's permission setting. Unlike other relational database management systems (RDBMS) such as Oracle, SQLite under HTML5 does not have GRANT operation to limit access to particular users for a particular table. According to Newman (2005), a user who has read access also has access to the entire database. Likewise, if the user has write access, he/she also has write access for all tables in the database. The last disadvantage of SQLite is the file-locking issues. Newman (2005) does not recommend SQLite for high concurrency database where users might frequently use the database. With the

nature of SQLite as a single file, it will lock all requests until a previous writing process is finished. For a single-user smart phone, this should not be a problem since only one user will write data to the database at one time. On the other hand, if the system has to wait for data from different tables to be written one after another, this introduces usability issues (i.e., longer waiting time) especially for a large file. A solution to such the problem is proposed in database implementation section (Section 4.1.2).

According to Golding (2008), XML is used for data interchange between applications. XML is used as a web standard database to transport data between servers and between server and client. It is applied to overcome the challenge of data exchange from a web service to different mobile devices. XML is an open web standard to describe data independent of any particular operating system, computer language, protocol, or network (Golding, 2008). Consequently, XML can be used to integrate data from multiple sources. Another advantage of XML is that it can be styled and transformed before it is rendered with Extensible Stylesheet Language (XSL) as suggested by Gehani (2002). Golding (2008), commenting on J2EE presentation layer, introduces the concept of utilizing XML and XSL transformation process to adapt final output content for each particular device.

Despite all the above-mentioned advantages, XML also has some disadvantages including the speed of retrieving data, which is slower than SQLite, since XML has to be parsed first compared to SQLite. The traditional SQL data query is faster because it utilizes SQL statements and it has RDBMS structure.

MySQL version 5.4 has the advantage of SPATIAL data storage and feature extensions. Not all SPATIAL database features are included in MySQL standard package. Hence, an additional package is required to be installed to take advantage of SPATIAL Geographic library features such as distance and point. With the spatial database, MySQL version 5.4 allows the data to be stored, retrieved, queried, and manipulated as spatial data type with spatial index, but it does not come with Geographic library (Gehani, 2007). However, the library is still in a beta version to date. The spatial data table is only available with MyISAM database engine. However, the engine does not allow foreign key constraints like INNODB engine does. It does not support transactions (Gehani, 2007). According to Gehani (2007), the spatial database geometry class includes the point and distance procedure where they can be used to find a distance between two points.

3.6. Data Transfer

The REST web service offers many advantages compared to Simple Object Access Protocol (SOAP) web service (Richards, 2006). REST can be a solution to many mobile device web limitations. Amazon, EBay, and Yahoo are the examples of REST web services. The REST service architecture includes XML, HTTP, URI, and MIME types.

The first thing to consider when creating a REST is the URI. Unlike URL, URI is suitable for REST since it points to a resource of a web service and hence does not change over time. Richards (2006) suggests a structure of URI for a web service.

The HTTP methods that are commonly used for REST are GET, HEAD, POST, PUT, and DELETE (Create, Retrieve, Update, and Delete). Richards (2006) states that GET should only be used for retrieving a resource representation. In contrast, POST can be used for other operations rather than resource retrieval including resource creation, modification, addition, and deletion.

For security purposes, as a result of performing a GET request, according to Gregorio (2004) and Richards (2006), there should not be any side effects that users are unaware of, and, therefore, the implementation of a GET method should be safe and idempotent. An idempotent method provides the same result every time a service is requested.

Apart from URI, data format, and methods, we also have to consider the other types of web service status codes (Gregorio, 2004) – 2xx for success, 3xx for redirection, and 4xx for errors; which are used for checking REST return request status.

3.7. Short Messaging Service (SMS) Notification

Ushahidi (2009) utilizes FrontlineSMS text messaging system for their web-based handheld device information distributing during a disaster by user generated content (http://www.ushahidi.com). The free open source SMS messaging system is designed specifically for non-governmental organizations (NGOs). FrontlineSMS is also being used by a mobile health system called FrontlineSMS:Medic (Addo, 2009). According to Kiwanja.net (2009), FrontlineSMS uses a personal computer or a laptop and modems as a hub to distribute the SMS to registered mobile phones. Likewise, a mobile GSM phone in a network is ported to the system by sending a normal standard message that comes with the mobile phone to the system. Rochester Institute of Technology (RIT) also has a SMS emergency notification – Emergency Mass Notification System (EMNS) where the RIT community can register to the system via RIT information access centre.

4. Implementation

This project investigates mobile web resource tracking system solutions using mobile devices to be deployed during an emergency situation. The system provides real time data to a decision maker so the user can effectively and efficiently monitor resources and assess the situation accordingly. The system consists of two main features:

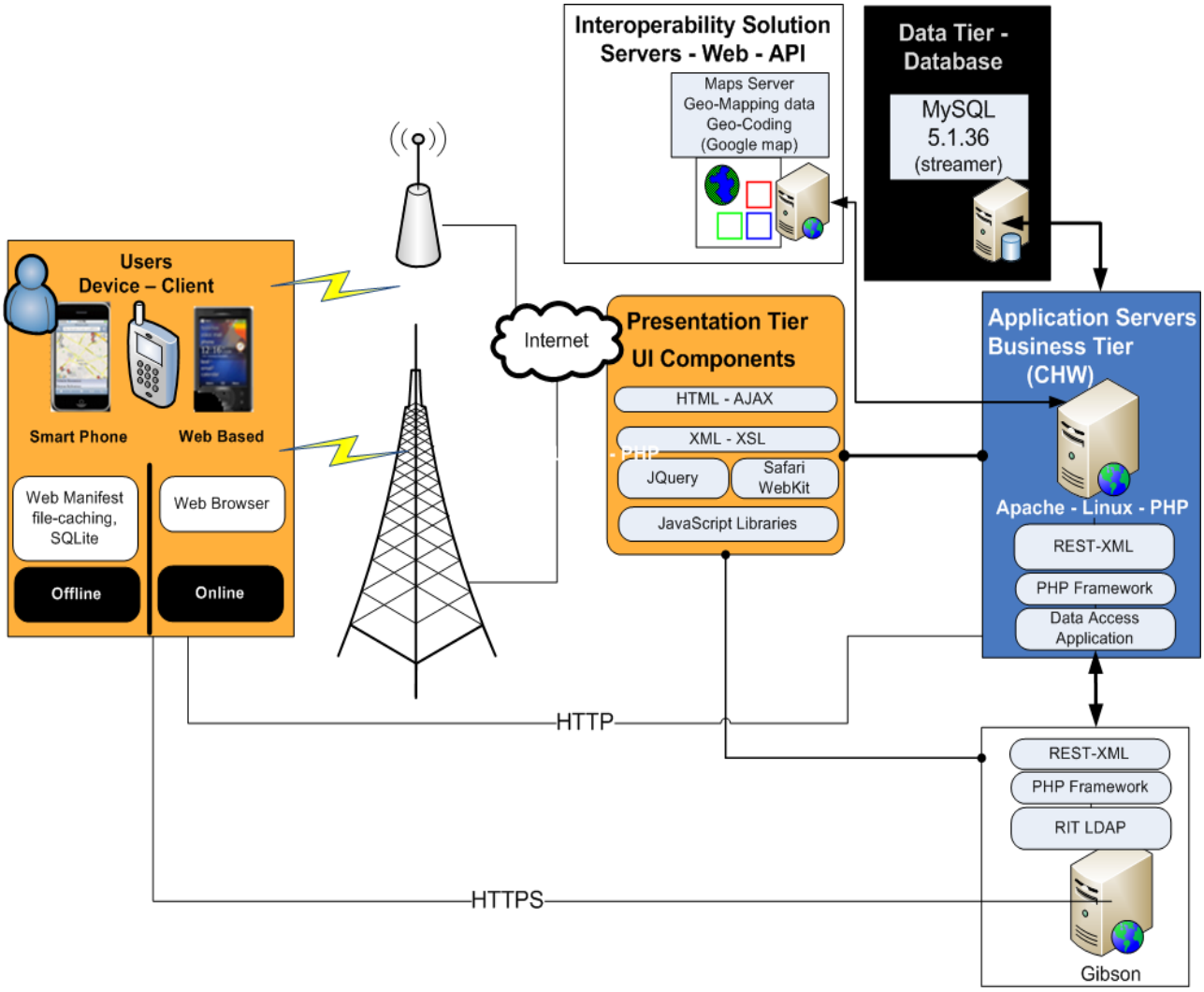
- Resource Position Tracking (e.g., Tracking geographical location)
- Resource Information Listing (e.g., Room Chemical Listing)

Mobile devices and smart phones provide convenient and location-independent usability. Web technologies have been selected to support cross-platform technology solutions appropriate for different mobile devices. Additionally, the system allows emergency team members to get assess to key data from any location, and to receive continual updates of the situations in the field. This allows the team to move quickly as the situation evolves and to reduce the amount of time waiting on a response for information requests.

According to Furht and Ilyas (2003), there are generally six groups of mobile devices which are web phone, wireless handheld devices, two-way pagers, voice portals, communication appliances, and web PCs. Mobile devices are technology dependent; each branch has its own standard and operating system. While Nokia and Sony Ericsson phones are run by Symbian OS, some other devices are built based on J2ME (Java for mobile phone). Other technologies include Pocket PC, Opera Software A/S, Mobile Linux, and Palm OS (Hillborg, 2002). Nonetheless, any smart phone with integrated web browser can access web services. Hence, this project is fundamentally focused on the Web technology and aims to transfer data to users via mobile web using Representational State Transfer (REST). A cellular network can be a bridge connection to the Internet during a disaster situation. The cellular network should provide a fast and reliable

connection. The WWW is utilized as a common technology for mobile devices that have different platforms. The system uses an interoperability solution, which utilizes client, servers, open source, and available Web API (as can be seen in Figure 11). It consists of three main layers – presentation tier, business tier or application tier, and data tier or DBMS system. The project cycle went through an incremental and changed process. For instance, the weather update feature, as part of initial design, was removed from the map application and the chemical listing feature was added instead as a result of usability testing during imagine RIT and after discussion with RIT public safety (as discussed previously in Section 3.1).

Figure 11: System Architect Overview



4.1. System Module and Installation

A list of technologies and the role of the technology used are as follows:

The system consists of the following primary components – database server, web application server, and client devices.

- The open source database management system (DBMS) - MySQL on a server called "STREAMER".
- The open source web server - Apache. There are two main web servers used for this system - the centre of handheld web server (CHW) and the RIT Gibson web server. The department of Information Technology student server (NOVA) is also used for deployment investigation and testing.
- Testing devices including a personal computer, a second-generation iPhone, and an iPod-Touch.

To implement the map feature, Google map API is used.

- Third party API for the mapping tool - Google map and some third party Java Script library for the map module.

In addition to the three-tier components and the third-party map API, the following are the software technologies used:

- Third-party JavaScript library for the map module was used for a marker multiple tab features and the labeling marker feature.
- Third-party library (e.g., JQuery) was used for asynchronous user and system interactions.
- PHP environment for the Apache platform and some third-party PHP libraries were utilized.
- Web Services (REST) was used for cross-domain data exchange in XML format.

- CSS, Extensible Style Sheet Language Transformation (XSLT), Extensible Style Sheet (XSL) and XML technologies are combined to create a friendly user interface in the chemical listing model.
- Encryption and security techniques.

4.1.1. Web Server

CHW is an Apache open-source HTTP web server. The map, the view online chemical listings page, and the user login authentication modules are hosted here. The entire database query and business tier are run on the CHW web server. The Gibson web server hosts the secure chemical listing cached off-line data. Still, Gibson uses CHW to communicate with the data tier using REST technology.

NOVA is used for testing and was investigated to host the chemical cached listing application. NOVA is protected by a very secure firewall. Unlike Gibson, NOVA is not connected directly to RIT LDAP, and it is protected by a Virtual Private Network (VPN) firewall which only allows access through the RIT network. The VPN feature like NOVA was recommended by Golding (2008) as a solution to enable end-to-end cellular network encryption across the Internet. However, it is a closed system, where only the network inside RIT and RIT authorized users can access the system. The Gibson web server was selected as a final deployment server since it is HTTPS enabled and it can be accessed by users outside of the RIT system. Here is a list of the advantages of using the Gibson web server:

- Provides HTTP Secured (HTTPS) protocol – SSL solution
- Connects directly to RIT LDAP server authentication
- It is protected through the RIT firewall
- It is maintained by the RIT security team, ITS Enterprise web applications

The RIT Gibson web server is selected to host and transfer data to be stored on a client device. Like the NOVA server, Gibson is an Apache HTTP web server. Gibson has the advantage of being connected directly to RIT HTTP authentication using LDAP. Therefore, the system can be configured to allow only RIT authenticated users to access it. RIT has information security policies which mandate and enforce the use of encryption when passing RIT computer account identifications. Therefore, Gibson web server and RIT LDAP server authentication settings are used to protect the important chemical- listing viewing application where the data is cached for off-line purposes. Thus the data transferred from Gibson to a client device is transferred through HTTP Secured (HTTPS) protocol. The HTTPS protocol utilizes SSL technology to protect data being transferred over the network as discussed earlier in the security section. The HTTPS protocol is a suggested technology by Golding (2008) for the encryption of data as protection in an end-to-end mobile web network. Also, SSL is the secure protocol normally used in credit card encryption. In addition to the secure protocol with LDAP authentication, the Gibson web server has a team of RIT security support that already oversees the security issues of the system as a whole.

4.1.2. Database Implementation

The three main databases used in this system are SQLite, XML, and MySQL. SQLite is selected to store off-line chemical list data. From the research as discussed in the previous section, SQLite has the disadvantage of file locking for the writing operation. Erl (2009) suggests that data replication should be used when necessary to spread out the user data access and hence improve system response time. Consequently, two databases were created to store two different tables – room data and chemicals stored in various rooms. In addition to the file locking issue, two SQLite databases were created to overcome the 6 MB limit size issue in the future. Nevertheless, currently data is only stored in one database since the size of the data is less than 6 MB. XML file format is utilized to store data for transferring between servers and between server and client. Golding (2008) states the challenges of using HTTP over a wireless link is that the network latency can be a problem in terms of loading and viewing a page. Golding (2008) demonstrates that the network latency of Radio Frequency (RF) can become a major issue compared to the actual time that is needed to spend in sending data. He believes that the actual amount of data being fetched is not as important as adding delay time if many requests are made. Apple engineers also suggest transferring data only once to save the battery of iPhone or iPod Touch (Geddis, Adler, & Brouwer, 2009). As a result, the chemical list XML file is generated to have the entire chemicals on a floor and the data is stored in one XML file. Data is stored and pulled from the MySQL database management system. The database relational diagram can be seen in the ERD diagram to retain data on MySQL database management system are divided in two main groups – chemical listing data and user data. Chemical listing data part consists of room, building, and chemical tables. The user database contains all user data including the user login data with the related building and room data. Despite the fact that MySQL version 5.4 has

the advantage of SPATIAL data storage and features, it is still a beta version. Hence, MySQL (on Streamer) version 5.1.36 is used instead.

The entity relationship diagram can be seen in Appendix 1. This is the redesign to address the system according to incremental changes towards the end of the project. The database contains two main parts – user data (i.e., resource people) and chemical data. The user database part is designed to keep track of a resource person in an emergency response team. RIT campus is used as an example for an emergency location. The resource current geo-location, assigned responsibility, contact information, and user authentication login session are parts of the system. The chemical database part is designed to represent potentially harmful chemicals in an emergency situation (e.g., fire). RIT location (i.e., a room in a building) is used as an example. The data is not a real data and is only used for the project demonstration. Below is the definition of resource tracking database entity:

- PERSON is a resource individual that can be either a doctor or a nurse or a firefighter or a police officer. Each person is identified by a personID. The PERSON record has first name, last name, date of birth, password, the account created date, account deleted date (if applicable), email address, and a role. The email address is the user name that the user uses to log into the system. The user account to log into the system is different from the account that is connected directly to RIT LDAP system. The RIT account is not recorded in this database.
- Each role is identified by a roleTypeID (ROLETYPE). A person can have one of four roles - doctor or nurse or firefighter or police officer. In the future, one or more roles can be added to the table. DOCTOR, NURSE, FIREFIGHTER, and POLICE are separated from their parents' table (i.e., PERSON) to allow frequent updates on the geo-location of

each resource person. A resource person who is a fire-fighter and a police officer would be more likely to have his or her location updated more frequently than a doctor or a nurse. Consequently, when a record is updated, we would query a smaller table compared to having only one Person table to update (i.e., performance). In addition, it allows table access to only the one that is required and hence it is a better security solution. However, the horizontal partition solution is the tradeoff of having the cognitive load by making four tables.

- A RESPONSIBILTY can either be assigned or not assigned (STATUSTYPE). A PERSON can have one or more assignments or not at all. The RESPONSIBILTY record has description, start date, and end date of an assignment.
- Each PERSON has longitude and latitude location data that are periodically updated. The update time is recorded in the updateDateTime attribute.
- Every PERSON has a UserSession data to authenticate his/her login. Each USERSESSION is identified by personID. For each login, the generated sessionID, tokenID, and the login time data are stored.
- Each PERSON can have phone contact information. The contact number can be one of the following types - an office phone number, a home phone, a mobile phone, or a pager number.
- The ADDRESS information includes zipcode, city name, state, street address, country. The address belongs to either a home address or an office address type.
- A BUILDING can have one or many addresses; every room can only be in one building.
- A PERSON can have the same address as the BUILDING that contains chemicals.
- buildingID is used to identify a building that contains a building number, a building name, and a building abbreviation. One BUILDING belongs to one ADDRESS.

- roomID is used to identify a ROOM in a building. Each room has a number.
- A ROOM might store one or many chemical substances or none at all. Each CHEMICAL is identified by chemicalID. A CHEMICAL has name, title, and link to an online MSDS database. The chemical name is the name of a chemical substance, while the chemical title is the name with coding according to the MSDS database. The CANOFFLINE attribute was added to determine if a chemical list is allowed to be displayed and stored locally or not.

This project cycle went through the process where the approach taken to develop the prototype was incremental. As a result, tables were changed and added as a feature was changed or a new feature was added. For instance, initially the chemical listing feature was not part of the project scope. However, after several meetings with tested users (RIT public safety), the functionality was recommended as an important part to the emergency response team. Consequently, the chemical related tables and attributes were added to the design. In addition, the database was demoralized as the prototype has evolved over time. For example, PERSON table was initially designed as a single table to contain PERSON data and PERSON geo-location. However, to improve performance four tables that represent each resource group (i.e., doctor, nurse, fire fighter, and police officer) were created. The table is used to record geo-location data as it is updated. Not all of the four resource groups would require having position updated frequently. For instance, police officers would move more often compared to doctors and nurses. Consequently, the police officer table would get updated more often than the doctor table. To sum up, the database design in this project could be reevaluated and the database refactoring technique could be studied to improve the design. Ambler (2003) states that the database refactoring technique is the technique to make change to a database schema while the behavioral and the informational semantics of the database are retained.

4.2. Applications

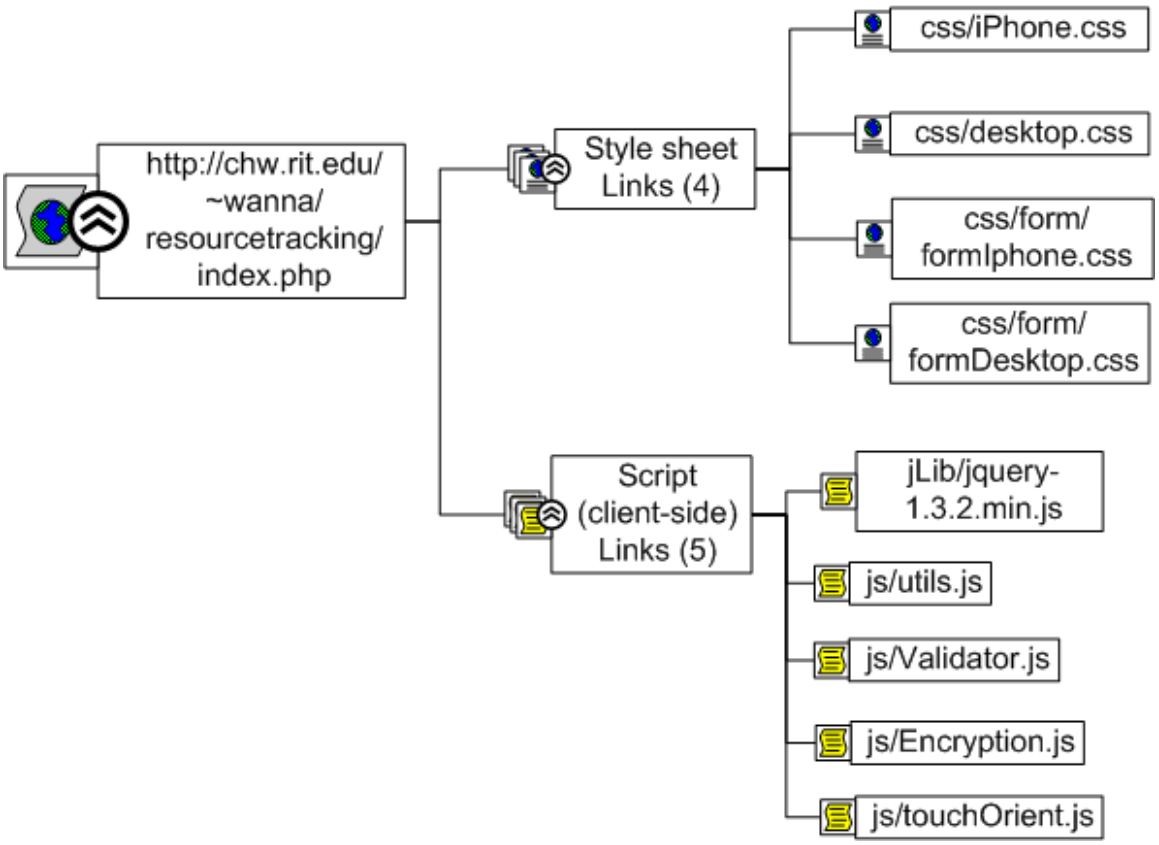
4.2.1. Login Authentication and Security

Table 1 shows the validation list for user authentication input form. The username and password inputs are validated before being transferred to the web server for authentication checking. If the password passes, the client-side validation is then encrypted and sent to the server. The login file map can be seen in Figure 12.

Table 2: Login Validation Check List for username and password form field

Check For	Function Name	Character Check	Action
Encode input before validate input to prevent escape characters	Validator.removeSpaces (stringinput)		
Empty String or NULL string	Validator.Validate function(selector, inputType)	· Empty string · “\0” · “%00” · \\00 · 0 · '0' · null · false · undefined	Not valid Validator.invalid = function()
Space between characters		Empty string	Remove space Validator.removeSpaces (stringinput)
Email	Validator.validateEmail()		

Figure 12: Login file structure (index.php)



As discussed in the security section, the password is stored in the database as an encrypted password. The input password is also encrypted using SHA-1 algorithm before being transmitted to the web server. SHA-1 is selected for the user authentication application because it is a one-way encryption where there is no need to decrypt the password. In addition, SHA-1 requires less computational time when compared to the HMAC algorithm and it is more secure than MD5. With the tradeoff of usability and security, the user input password is encrypted using JavaScript. This raises the question of how secure the algorithm could be when the code can be viewed through web browsers on normal personal computers. The solution for the issue is to have the encrypted JavaScript class file encoded and optimized through code-space removal where the file becomes non-readable. Consequently, the code or the algorithm is protected. However, the

downside of this technique is that, for encoding the whole file, the normal 5,485 bytes "Encrypted.js" file is equivalent to 11,637 bytes when the file is encoded. This introduces a high overhead for the system that has direct impact on the page respond time and the application performance. Therefore, the solution was not selected. Nevertheless, the current one-way encryption method is already applied for the security.

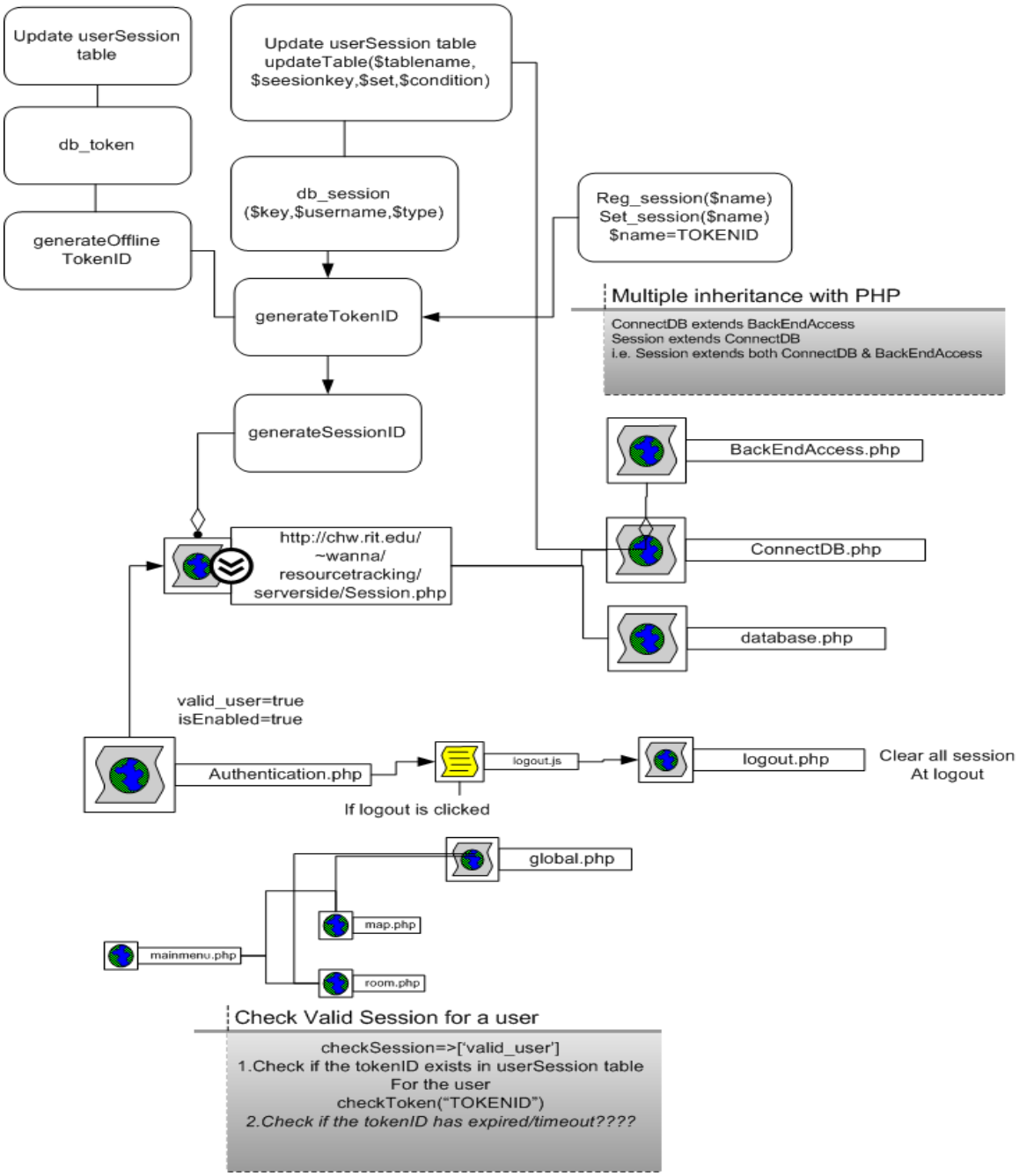
The JavaScript encryption code is based on the PHP SHA-1 code. The encryption code is taken from PHP.JS (http://phpjs.org/functions/sha1) by Kevin van Zonneveld (2009). Ballad and Ballad (2009), and Kim and Skoudis (2009) suggest using a well-known, carefully developed, and tested encryption code by others. Consequently, the code taken is fitted in the category.

After receiving the validated username and password from a client, the web server then queries for corresponding data from the database server (i.e., searching for matched user and password). If the user is a legitimate user, then he/she is authenticated and allowed to access the rest of the application.

Both the server-side authentication and the Gibson or NOVA firewall protect the application while the network connectivity is active for the chemical list HTTPS request mode. However, the protections are no longer valid after users accept the caching mode of the manifest file. The manifest file is used for storing Web files (i.e., HTML, JavaScript, CSS, and image files) in application cache (http://developer.apple.com/safari/library/documentation/iPhone/Conceptual/SafariJSDatabaseGuide/OfflineApplicationCache/OfflineApplicationCache.html, 2009). The issue and a proposed solution are addressed in the data availability for off-line mode section (5.3). To protect against

session hijacking, a random generated one-time token key is created. All sessions are destroyed after users log out (Ballad & Ballad, 2009; Nexus, 2007). *Figure 13* shows the login authentication program flow.

Figure 13: Authentication Session/Token key check at the server side



4.2.2. Content Presentation for Chemical Listing Using XSL and XSLT Display Transformation

According to Rischpater (2001), the combination of XSL processor, XSL style sheet, and XML content is an abstract markup language based on XML document. Consequently, it is said to be free of device constraints where different web browser clients support the use of the technologies. Therefore, Rischpater (2001) concludes that the advantage of utilizing the abstract markup language for mobile device screens will help scalability in deploying. By adding additional style sheets, mobile web application content can be presented differently on different mobile web clients in the future (e.g., different screen sizes). Figure 14 shows the file structure of the chemical listing page. The web content presentation program flow is shown in Figure 15.

Figure 14: Room Web Map

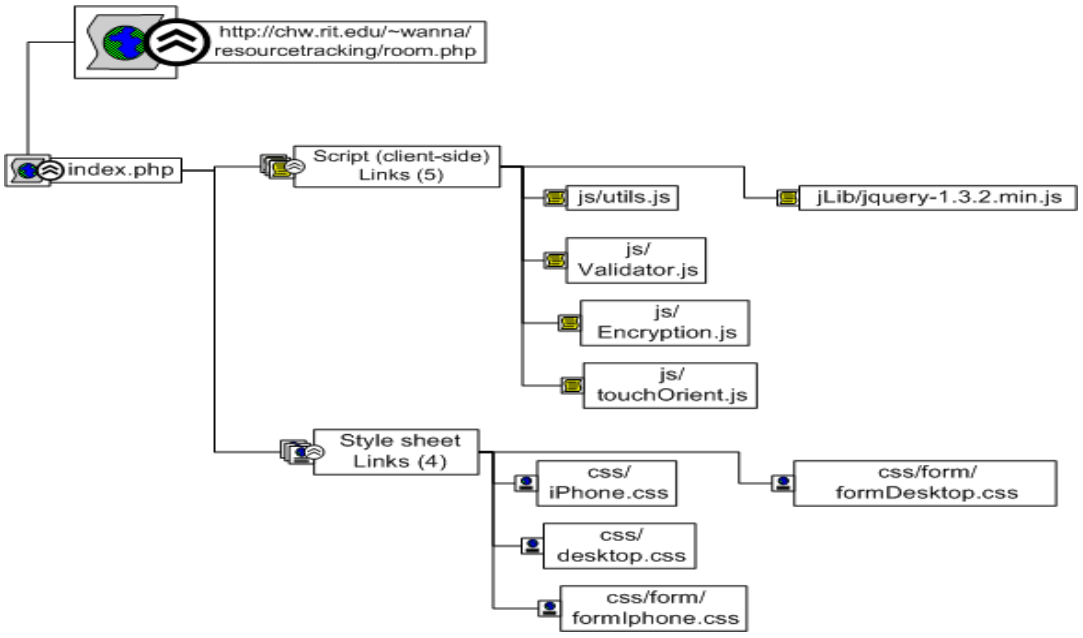
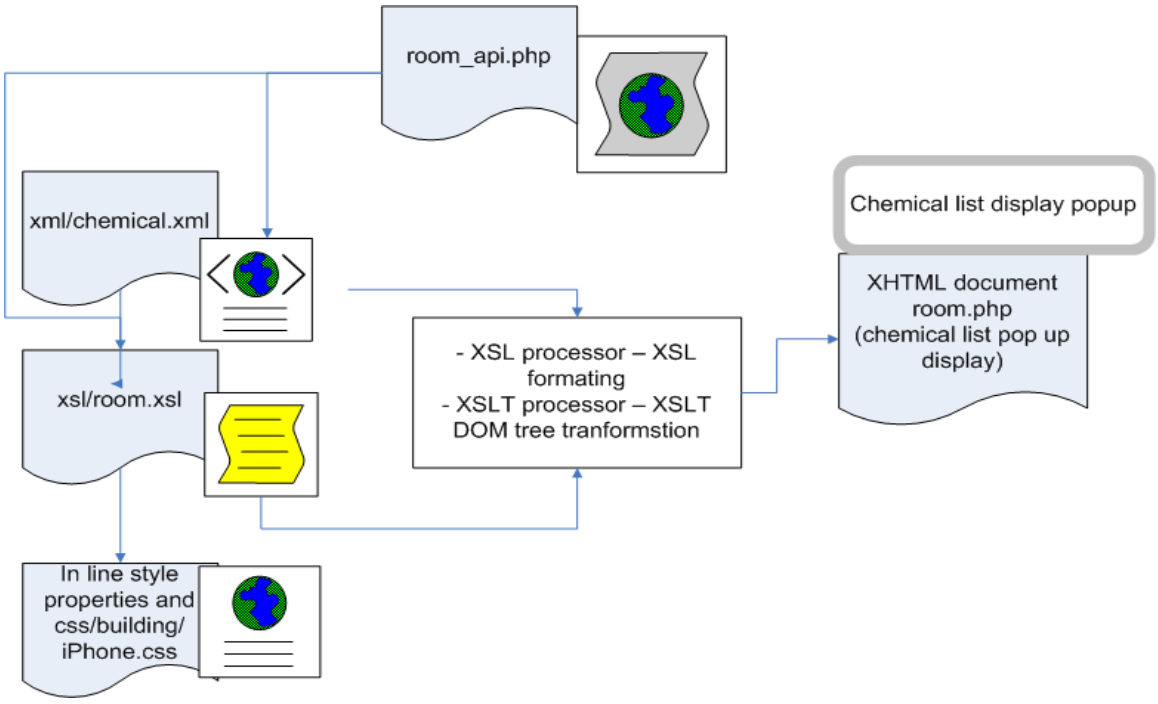


Figure 15: XSL web content presentation flow for the room chemical listing



A room contains one or more chemicals. The chemical has a link to the online Material Safety Data Sheet (MSDS) database (http://www2.hazard.com/msds/gn.cgi?whole=whole&start=0&query=). The link of a chemical list is pulled from the database through PHP file as shown in Figure 16. Figure *17* shows a chemical listing in a form of XML representation (chemicalist.xml).

Figure 16: PHP code to get search result to link to MSDS database of a chemical

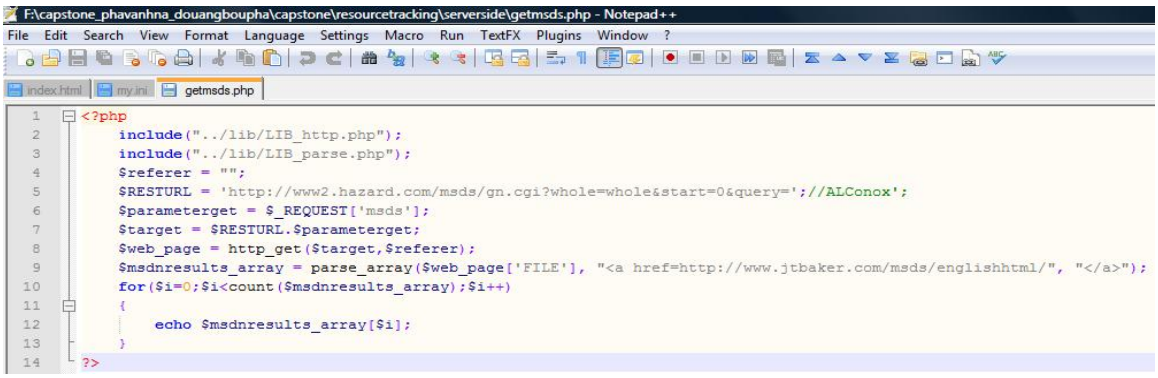
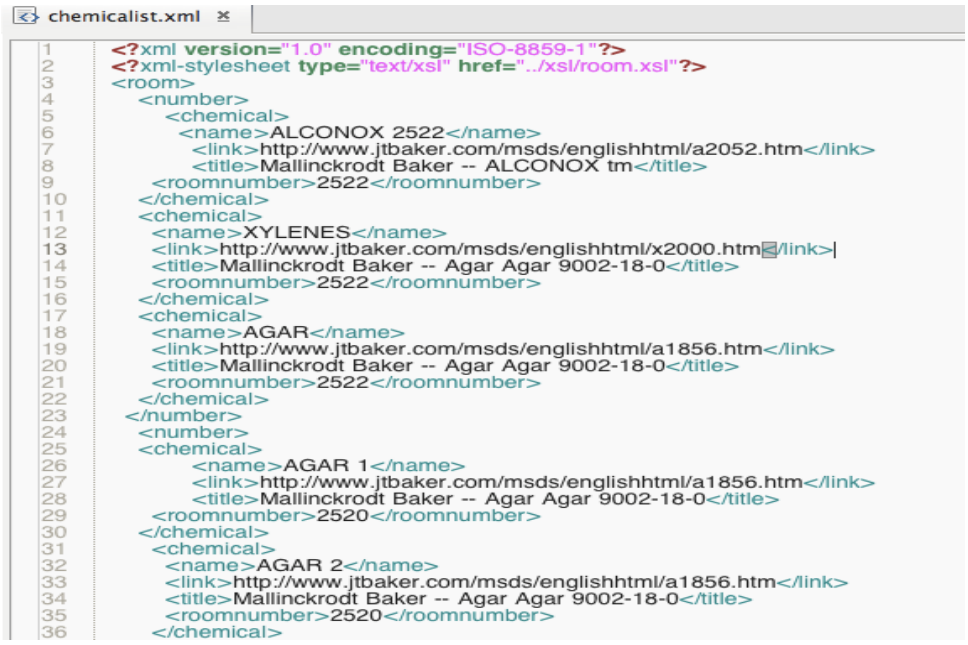


Figure 17: Chemical List XML content



```
chemicalist.xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="../xsl/room.xsl"?>
<room>
   <number>
      <chemical>
        <name>ALCONOX 2522</name>
          <link>http://www.jtbaker.com/msds/englishhtml/a2052.htm</link>
          <title>Mallinckrodt Baker -- ALCONOX tm</title>
      <roomnumber>2522</roomnumber>
    </chemical>
    <chemical>
      <name>XYLENES</name>
      <link>http://www.jtbaker.com/msds/englishhtml/x2000.htm</link>
      <title>Mallinckrodt Baker -- Agar Agar 9002-18-0</title>
      <roomnumber>2522</roomnumber>
    </chemical>
    <chemical>
      <name>AGAR</name>
      <link>http://www.jtbaker.com/msds/englishhtml/a1856.htm</link>
      <title>Mallinckrodt Baker -- Agar Agar 9002-18-0</title>
      <roomnumber>2522</roomnumber>
    </chemical>
  </number>
    <number>
    <chemical>
          <name>AGAR 1</name>
          <link>http://www.jtbaker.com/msds/englishhtml/a1856.htm</link>
          <title>Mallinckrodt Baker -- Agar Agar 9002-18-0</title>
      <roomnumber>2520</roomnumber>
    </chemical>
      <chemical>
        <name>AGAR 2</name>
        <link>http://www.jtbaker.com/msds/englishhtml/a1856.htm</link>
        <title>Mallinckrodt Baker -- Agar Agar 9002-18-0</title>
        <roomnumber>2520</roomnumber>
      </chemical>
```

XSLT style sheet named “room.xsl” is written to transform the XML DOM content representation into a specified format for display while applying CSS style sheet and in-line styling properties. The result is the XSLT processor generates markup for display on users’ mobile web browser screens. XSLT technology is selected because it is a related W3C specification for rendering XML documents to a display format. XML document is a web standard language for displaying content on mobile devices (Pashtan, 2005). In addition to the web content display format, XSL is also the code used as the processing control to pick specific elements within the XML document. Hence, it is a way for optimization by replacing the functionality that is normally performed with JavaScript. Consequently, it can help reduce the JavaScript code size on the client. For the chemical listing, the conditional processing control is used to display only four (4) items at a time. The control code is used to count the total number of chemicals in the list and to keep track of how many more are left on the list in the room to be

loaded for the display. A <sort> element by a chemical name is added to display the list sorted in ascending order (room.xsl).

4.2.3. Data Availability for Off-Line Mode

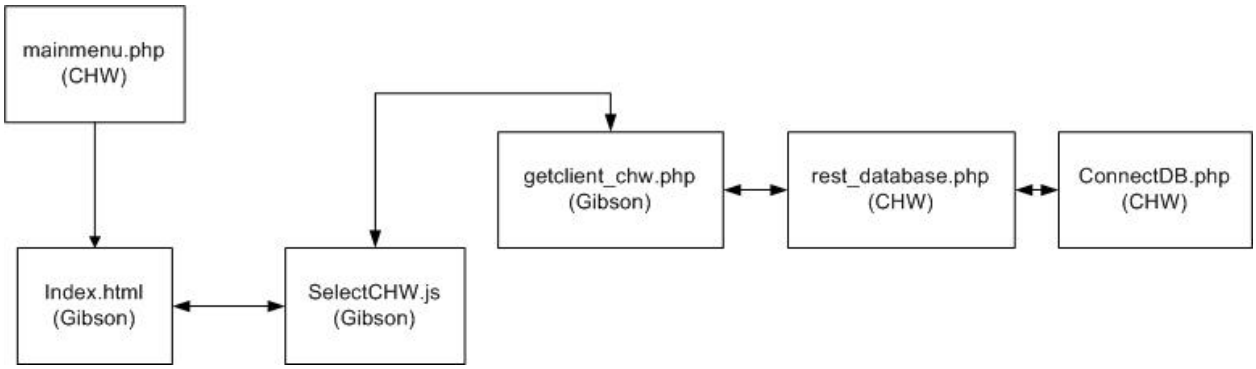
The two important aspects of the project are mobility and the urgent need for data in emergency situations. The mobility allows the users to assess data from anywhere at any convenient time. During an emergency situation, it is critical that a user is able to locate important data when needed. However, during such situations there is a risk of heavy network traffic or worse no network connection at all. Hence, data persistence and data availability are critical during such situations. One solution is to have off-line data available using browser technology where the data is cached. Consequently, the caching data can be viewed even though the connection is lost.

After research, the browser-manifest file caching method and storing of selected data in SQLite techniques are selected. However, due to security reasons, each mobile device is set through a factory setting to have limited space available for manifest file caching and the SQLite database. This space limit is overcome by saving only selected data and by using optimization. The manifest file application cache is supported by Firefox 3* and 3.1, Safari 3* and 3.1, and iPhone OS 2.1+.

In addition to the space issues, there are also security concerns. Therefore, the chemical listing room cache mode is implemented on Gibson whereas the view-only mode module is implemented on CHW. Gibson is selected to be used in the view and save mode because of security. Gibson already has a server and network security set so that only users inside the RIT network can access the server. As a result, only those who have access to RIT can see the view

and save mode to have off-line data available. Nevertheless, if the device is stolen, anyone can easily view the off-line cache page. To overcome such issue, a randomly generated token key is created each time the user views the page. The view-and-save mode is protected by authentication using the random generated token key. Figure 18 represents the program flow between Client, Gibson web server, and CHW web server for retrieving chemical listing data of a room.

Figure 18: Retrieving Room data from MySQL and store on SQLite via CHW web server from Gibson (program flow chart)



Connectivity status of the network availability (i.e., with network or without network connection) is checked asynchronously using AJAX technique. Figure 19 shows the code implemented for this feature. The code is called from the chemical list view page on Gibson. The program flows can be seen in Figure 20. To enable manifest cache file the add type cache manifest is added to the .htaccess file - "AddType text/cache-manifest .manifest". Also, the cache files for the off-line mode, in particular the SQLite client-side database file, are stored with CACHE MANIFEST in the "viewsave.manifest" file:

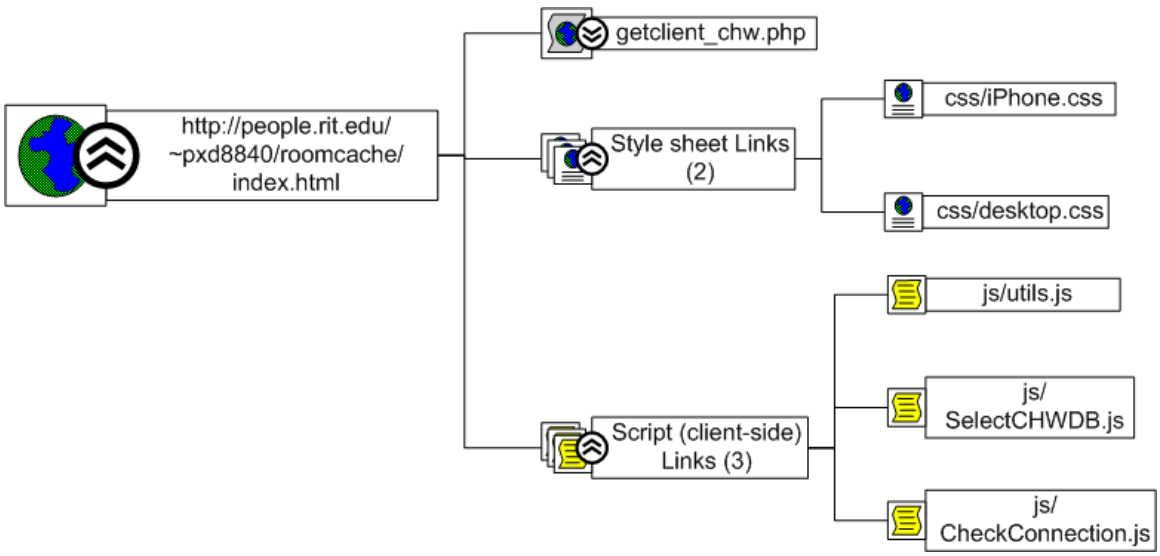
- js/_cachestatus.js (Ryan., 2009)
- js/utils.js
- js/CheckConnection.js

- js/SelectCHWDB.js
- index.html
- css/iPhone.css

Figure 19: JavaScript code to check for network connectivity status

```
room.php   CheckConnection.js
?id /*/ for f() if try
var CheckConnection = new Object();
CheckConnection.updateOnlineStatus = function(msgconnection)
{
   var connection = navigator.onLine ? "ONLINE" : "OFFLINE";
   var status = gebid("status");
   var state = gebid("state");
   status.setAttribute("class", connection);
   state.innerHTML = connection;
}
CheckConnection.loadConnectStatus = function()
{
    var d = document;
    CheckConnection.updateOnlineStatus("load");
    d.body.addEventListener("offline", function () {CheckConnection.updateOnlineStatus("offline")}, false);
    d.body.addEventListener("online",  function () {CheckConnection.updateOnlineStatus("online")}, false);
}
```

Figure 20: Off-line data cache web map for chemical listing



As the data is being transmitted, it should first encrypt using the hashing method (e.g., HMAC). The encrypted data will then be sent from the server via REST service. However, decrypting the data with JavaScript on the client side can take a lot of time and hence this method was not used. The HTTPS server (i.e., Gibson RIT web server) is used instead.

The off-line chemical list web application is protected by server side authentication. In addition to the set server side authentication, it is also protected through the firewall of RIT network (i.e., Gibson server). Consequently, only authorized RIT users are allowed to connect to the system, and the RIT wireless network connection is encrypted using WPA where transmitted data is encrypted. Figure 21 shows the .htaccess code written for the feature on Gibson.

Figure 21: Server-side LDAP authentication using htaccess

```
AuthType Basic
AuthName "RIT"
AuthBasicProvider ldap
SSLRequireSSL
AuthLDAPUrl ldaps://ldap.rit.edu/ou=people,dc=rit,dc=edu?uid?sub
AuthzLDAPAuthoritative off
require valid-user
```

Table 3: Off-line data persistent solution and its trade-offs

Problem	Solution	Impacts
· Data persistent during no connection time · Performance during peak time	· Web caching · Storing data using JavaScript SQLite database (Maximum size 6 MB)	· Limited file size that can be stored on the device · Increases risk of data security
· File size problem	· Optimization · Only store selected data	· Usability · Raises security concern
· Security	· Cache files can only be retrieved from trusted network (Gibson) – HTTPS · Only store non-sensitive data locally and sensitive data on the server · Using RIT network authentication to protect the file – LDAP	· Reduce security risk from the unauthorized users
· Data security on lost/stolen device	· Utilizing data encryption for chemical listing in the SQLite database can be implemented in the future	· It can affect system performance due to longer computation time
· Data security on lost/stolen device	· User authentication on one time use random generated token id. The token id can be used as a public key to encrypt and decrypt data in the SQLite database in the future	· It will increase the size of file that could reach the file size limit · It will affect performance – due to computation time · It has the impact on usability
· Data security on lost/stolen device	· Remotely delete data (can be implemented in the future)	· Timing - it can be too late by the time the device is identified to be missing · Network Connection – it requires a device to be connected to Internet to have data remotely wiped for iPhone 3GS

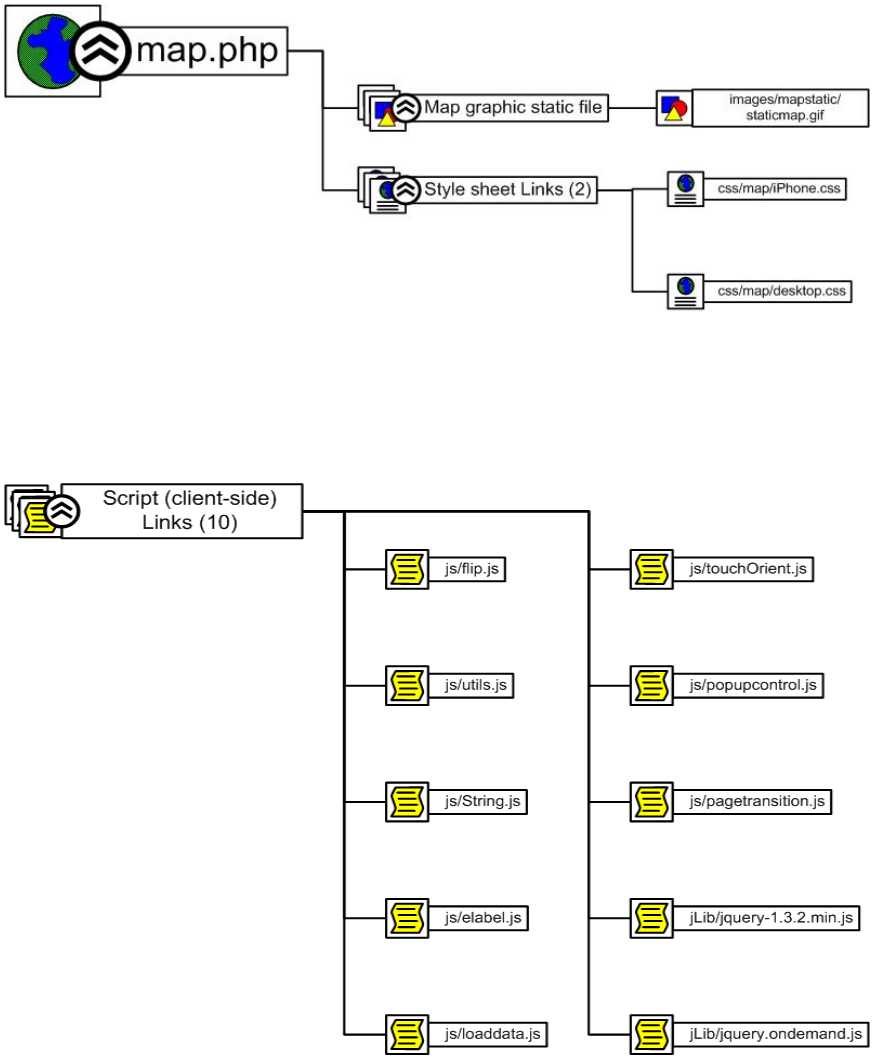
4.2.4. Location Tracking

To track the real time position of a moving resource, the resource must have the GPS-enabled device. As stated in the review of the literature, GPS is being used worldwide by national mapping agencies for geographic information systems and for natural disaster prediction systems. Despite the power consumption disadvantages, GPS is still a reasonable choice for this system. GPS geo-location is retrieved from the iPhone GPS-enabled device. A native Object-C code is written to capture a device GPS location. In the case of an iPod Touch or a device that is not GPS-enabled, Wi-Fi positioninig service is used instead to capture a user location. This is done through a client-side coding (JavaScript) (http://developer.apple.com/safari/library/documentation/AppleApplications/Reference/SafariWebContent/GettingGeographicalLocations/GettingGeographicalLocations.html, 2009) as suggested by Jefferson (2009).

The system provides location data in geo-coding coordinates (i.e., latitude and longitude coordinates). The mapping data and other additional positioning information are retrieved from an online available map server (e.g., Google or Live search map). Since the overhead for Google map and the image files for icons on this page are already large, the XML current location file is not encrypted. This is a trade off between security and performance. If MySQL SPATIAL data features and indexes are available, this could potentially improve the performance of retrieving the geolocation data from the database. The map page is protected by user-login authentication. Figure 22 displays the map page file listing. The geolocation data is recorded in a XML format and is parsed by XMLHttpRequest. A map marker has tab content showing user's information.

"tabbedmaxcontent.js" is a Google Maps API utility library by Liu (2009). "elable.js" is another Google Maps API by Williams (2009) to display and label each marker (i.e., icon) on the map.

Figure 22: Map page file structure



5. Conclusion and Recommendations for Future Work

One of the main objectives of this project is to provide a real time prototype of a resource-tracking solution that is usable during an emergency situation. The system is created to provide real-time tracking of resources, sending position information about the resources (including human resources) to a decision-maker so that managing the resources can be performed effectively and efficiently during a time-sensitive situation. The "chemical listing" feature is implemented to demonstrate how a mobile device application can be used to provide data as a safeguard for an emergency response team, and possibly to the general public, before they enter a questionable area. Consequently, this is a tool that allows public-safety teams to assess a situation in real-time. In addition, managing potentially risky issues (i.e., chemicals) when there is a disaster such as a fire is also important to a decision maker.

With Internet connectivity and mobile accessibility, the system offers the advantage of information availability anywhere on- or off-site. Another important focus of the project is the usability of web-based applications on mobile devices and ensuring suitability for a time-sensitive scenario. This project does not address the hardware challenges of mobile devices (e.g., battery life) and the wireless network connection issues directly. Instead, the system design and implementation were selected as the best compromise between four key factors – availability, performance, security, and usability.

There were many issues encountered in implementing this project. The scope of the project was not clear at the start. Instead of implementing just a prototype application, I researched factors and best solutions as more and more related issues evolved. The project went through an

incremental circle where features were changed and added. As a result, the project ended up covering different areas of the architecture, usability, availability, performance, and security. Consequently, it demonstrated the difficult process for a single developer to oversee the entire system (e.g., database design, architecture, security, usability) as it changed over time. For instance, the database structure was redesigned to include a new feature (i.e., chemical listing data). The database structure shown in Appendix 1 is not the actual current database. It is the redesign to anticipate the future improvement for user data (i.e., doctor, nurse, fire fighter, and police officer tables). The database refactoring technique should be further investigated and applied for future improvements. The proposed solutions, including the database design, could be changed according to the real application requirements.

As discussed, this project design depends on many trade-offs between security, availability and performance where usability is the most important factor. The current usability solutions do not accommodate color blindness. Selected colors to delineate rows in chemical listing table, for instance, should have a higher degree of contrast with each other (Hoffman, 1999). In addition, simple symbols may be used instead of colors to represent how dangerous each chemical is.

Another challenge for this project is to design the architecture and the selected the best technologies to incorporate the different parts of the system. REST was used for the purpose. Some of the implementation solutions rely on future improvements in existing technologies. For instance, the iPhone Map API was not initially available and was not released until towards the end of the project. Consequently, there was enough time to compare the Web solution and the iPhone native solution. Though, the Native code for the map feature was programmed; there was

not enough time to perform bench marking. The improvement of mobile web browser in the future would also be a factor in improving the system performance and user interface design.

Testing is another problem for this type of application, testing each feature was very time-consuming as both desktop platform and mobile platform were tested. The limited privileges available on NOVA server and Gibson server limit the ability of installing new libraries for the application such as the SPATIAL database features. Though, the full privilege on Streamer and CHW were granted later in the project, the servers also host other important data. Consequently, changing a configuration on the servers could potentially introduce a problem to the entire system. The Web security addressed should be taken into consideration, nevertheless moving all the applications to the HTTPS enabled system would help to increase system security.

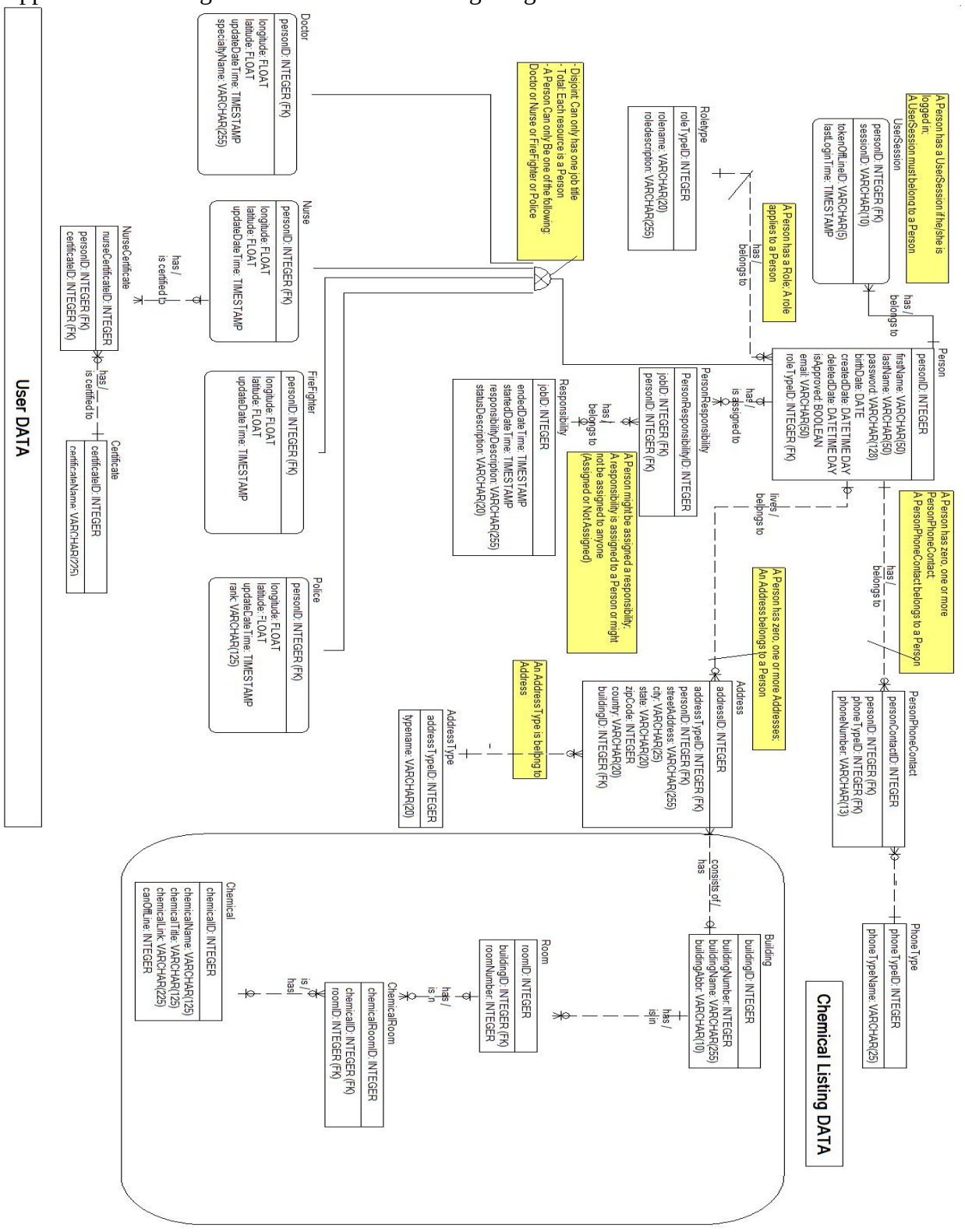
As the design relies upon pre-existing commodity products and services, research that addresses the significance of wireless network connectivity and mobile device battery life during a crisis situation could provide directions for improving the system. The following is a list of future works that could be further investigated to improve the system design and implementation:

- Utilizing and implementing unit automatic testing with the potential to speed up the development process. In addition, decreasing the chance of overlooking system bugs due to human error testing.
- Using real devices other than the iPhone to test the system on the web browser of each mobile platform.
- Performing real-time testing with emergency personnel.
- Implementing an offline version as a location-tracking module using an image map.
- Implementing search function for chemical data.

- Encryption implementation of data on SQLite database on the client side for the offline-caching mode on a device.
- Developing position tracking inside a building using wireless strength to indicate position. This could be developed using iPhone object-C API.
- Continuing on investigation of the use of SPATIAL database in different DBMS. Re-examining the database design.
- Implementing an alert system to notify users in the event of a crisis situation other than the use of the short messaging service (SMS) system. One possible solution could be to use XSLT technology to send a command to print the warning out to a printer.
- Setting up a team comprised of different disciplines to collaborate on the project instead of using a single developer method.

6. Appendices

Appendix 1: Redesign ERD Database Modeling Diagram



Appendix 2: Document Size

Figure 23: Document size for map page

Document Size - http://chw.rit.edu/~wanna/resourcetracking/map.php

Documents (1 file)	**9 KB**
http://chw.rit.edu/~wanna/resourcetracking/map.php	9 KB
Images (27 files)	**433 KB**
http://chw.rit.edu/~wanna/resourcetracking/images/icon/csg-icons.png	162 KB
http://chw.rit.edu/~wanna/resourcetracking/images/icon/building30x30select.png	32 KB
http://chw.rit.edu/~wanna/resourcetracking/images/icon/fireman30x30select.png	32 KB
http://chw.rit.edu/~wanna/resourcetracking/images/icon/police30x30select.png	31 KB
http://chw.rit.edu/~wanna/resourcetracking/images/icon/doctor30x30select.png	31 KB
http://chw.rit.edu/~wanna/resourcetracking/images/icon/nurse30x30select.png	30 KB
http://mt0.google.com/vt/lyrs=m@107&hl=en&src=api&x=4656&y=6014&z=14&s=Galile	13 KB
http://mt0.google.com/vt/lyrs=m@107&hl=en&src=api&x=4658&y=6015&z=14&s=Galil	11 KB
http://mt1.google.com/vt/lyrs=m@107&hl=en&src=api&x=4657&y=6016&z=14&s=Gal	11 KB
http://mt3.google.com/vt/lyrs=m@107&hl=en&src=api&x=4657&y=6015&z=14&s=Ga	11 KB
http://mt2.google.com/vt/lyrs=m@107&hl=en&src=api&x=4658&y=6014&z=14&s=Gali	11 KB
http://mt2.google.com/vt/lyrs=m@107&hl=en&src=api&x=4658&y=6016&z=14&s=Galile	10 KB
http://mt1.google.com/vt/lyrs=m@107&hl=en&src=api&x=4657&y=6014&z=14&s=G	9 KB
http://mt1.google.com/vt/lyrs=m@107&hl=en&src=api&x=4655&y=6015&z=14&s=Gali	9 KB
http://mt2.google.com/vt/lyrs=m@107&hl=en&src=api&x=4656&y=6015&z=14&s=Galileo	8 KB
http://mt3.google.com/vt/lyrs=m@107&hl=en&src=api&x=4655&y=6014&z=14&s=Gal	5 KB
http://mt3.google.com/vt/lyrs=m@107&hl=en&src=api&x=4655&y=6016&z=14&s=Galil	5 KB
http://mt0.google.com/vt/lyrs=m@107&hl=en&src=api&x=4656&y=6016&z=14&s=	4 KB
http://maps.gstatic.com/intl/en_us/mapfiles/poweredby.png	4 KB
http://chw.rit.edu/~wanna/resourcetracking/images/icon/fireman30x30select.gif	2 KB
http://chw.rit.edu/~wanna/resourcetracking/images/icon/police30x30select.gif	2 KB
http://chw.rit.edu/~wanna/resourcetracking/images/icon/nurse30x30select.gif	674 bytes
http://maps.gstatic.com/intl/en_us/mapfiles/shadow50.png	665 bytes
http://chw.rit.edu/~wanna/resourcetracking/images/icon/doctor30x30select.gif	649 bytes
http://maps.gstatic.com/intl/en_us/mapfiles/smc.png	389 bytes
http://maps.gstatic.com/intl/en_us/mapfiles/markerTransparent.png	260 bytes
http://maps.gstatic.com/intl/en_us/mapfiles/transparent.png	95 bytes
Objects (0 files)	
Scripts (15 files)	**193 KB (340 KB uncompressed)**
http://maps.gstatic.com/intl/en_us/mapfiles/164e/maps2.api/main.js	64 KB (177 KB uncompressed)
http://chw.rit.edu/~wanna/resourcetracking/jLib/jquery-1.3.2.min.js	56 KB
http://chw.rit.edu/~wanna/resourcetracking/js/tabbedmaxcontent.js	16 KB
http://maps.gstatic.com/cat_js/intl/en_us/mapfiles/164e/maps2.api/%7Bmod_jslinker,mod_stats,mod_api_gc,mod_drag,mod_ctrapi%7D.js	14 KB (38 KB uncompressed)
http://chw.rit.edu/~wanna/resourcetracking/js/map.js	13 KB
http://chw.rit.edu/~wanna/resourcetracking/js/utils.js	9 KB
http://chw.rit.edu/~wanna/resourcetracking/js/elabel.js	5 KB
http://maps.google.com/maps?file=api&v=2&sensor=false&key=ABQIAAAAm0OkW_th2FbCv4YYiCdtPRQcNyMxYsE8hrGuCzdvTUhAgwNXLRT5zOFc0XHu9VTdJbf-UOTHVFwz8w&maptype=mobile&async=2&callback=initMap	5 KB (15 KB uncompressed)
http://chw.rit.edu/~wanna/resourcetracking/js/loaddata.js	4 KB
http://chw.rit.edu/~wanna/resourcetracking/js/touchOrient.js	4 KB
http://chw.rit.edu/~wanna/resourcetracking/jLib/jquery.ondemand.js	2 KB
http://chw.rit.edu/~wanna/resourcetracking/js/popupcontrol.js	805 bytes
http://chw.rit.edu/~wanna/resourcetracking/js/flip.js	737 bytes
http://chw.rit.edu/~wanna/resourcetracking/js/pagetransition.js	462 bytes
http://chw.rit.edu/~wanna/resourcetracking/js/String.js	166 bytes
Style Sheets (2 files)	**24 KB**
http://chw.rit.edu/~wanna/resourcetracking/css/map/iPhone.css	12 KB
http://chw.rit.edu/~wanna/resourcetracking/css/map/desktop.css	12 KB
Total	**660 KB (807 KB uncompressed)**

Figure 24: Document size for view only chemical listing page

Document Size - http://chw.rit.edu/~wanna/resourcetracking/room.php

Documents (1 file)	**5 KB**
http://chw.rit.edu/~wanna/resourcetracking/room.php	5 KB
Images (15 files)	**29 KB**
http://chw.rit.edu/~wanna/resourcetracking/images/building/West_Face.png	6 KB
http://chw.rit.edu/~wanna/resourcetracking/images/building/South_Face.png	5 KB
http://chw.rit.edu/~wanna/resourcetracking/images/building/North_Face.png	4 KB
http://chw.rit.edu/~wanna/resourcetracking/images/building/West_Face-15.png	3 KB
http://chw.rit.edu/~wanna/resourcetracking/images/building/NoRoom2.png	2 KB
http://chw.rit.edu/~wanna/resourcetracking/images/building/FloorMap_09.png	2 KB
http://chw.rit.edu/~wanna/resourcetracking/images/building/NoRoom1.png	1 KB
http://chw.rit.edu/~wanna/resourcetracking/images/building/2550.png	1 KB
http://chw.rit.edu/~wanna/resourcetracking/images/building/2522.png	1 KB
http://chw.rit.edu/~wanna/resourcetracking/images/building/2520.png	935 bytes
http://chw.rit.edu/~wanna/resourcetracking/images/building/2570.png	923 bytes
http://chw.rit.edu/~wanna/resourcetracking/images/building/debug.png	781 bytes
http://chw.rit.edu/~wanna/resourcetracking/images/building/Wall.png	509 bytes
http://chw.rit.edu/~wanna/resourcetracking/images/building/Wall-12.png	261 bytes
http://chw.rit.edu/~wanna/resourcetracking/images/building/spacer.gif	43 bytes
Objects (0 files)	
Scripts (6 files)	**71 KB**
http://chw.rit.edu/~wanna/resourcetracking/jLib/jquery-1.3.2.min.js	56 KB
http://chw.rit.edu/~wanna/resourcetracking/js/utils.js	9 KB
http://chw.rit.edu/~wanna/resourcetracking/js/control.js	3 KB
http://chw.rit.edu/~wanna/resourcetracking/js/_cachestatus.js	990 bytes
http://chw.rit.edu/~wanna/resourcetracking/js/Login.js	828 bytes
http://chw.rit.edu/~wanna/resourcetracking/js/CheckConnection.js	653 bytes
Style Sheets (2 files)	**4 KB**
http://chw.rit.edu/~wanna/resourcetracking/css/building/iPhone.css	4 KB
http://chw.rit.edu/~wanna/resourcetracking/css/building/style.css	437 bytes
Total	**109 KB**

Appendix 3: List of File

Table 4: List of Document File

File Name	Description	Note
Authentication.php	Check user authentication	CHW web server
BackEndAccess.php	Back end MySQL query class	CHW web server
ConnectDB.php	Back end MySQL connection class	CHW web server
DomAccess.php	XML parse class for DOM document	CHW web server
Floor plan	Floor plan map of RIT building 70	taken from http://www.ntid.rit.edu/cat/summit/pdf/FloorMap.pdf (RIT, 2008)
Floor plan image map	Floor plan image map for RIT building 70	taken from http://www.morethanmachine.com/iphonecourse/ (Douangboupha, Jett, Dangelantonio, Sanjanwala, & Thomas, 2009)
getclient_chw.php	make a REST request to CHW web server to retrieve data	Gibson web server
getCurrentLocations.php	generates geo-location XML data	CHW web server
getGoogleStaticMap.php	retrieves Google static map	CHW web server
getmsds.php	retrieve link from online MSDS database	CHW web server
global.php	re-directs user on fail log in authentication	CHW web server
index.html	chemical listing page for off-line viewing	Gibson web server
index.php	login authentication	CHW web server
logout.php	clear session on log out	CHW web server
mainmenu.php	main menu listing three main pages - map page, chemical listing view only, and chemical listing with caching	CHW web server
map.php	Resource Location tracking	CHW web server
rest_database.php	Receive REST request and response to a request in XML format	CHW web server
room.php	chemical listing page for online viewing only	CHW web server
room_api.php	parses XML chemical data and XSL functions	CHW web server
Session.php	Generate Session/token key randomly	CHW web server

Table 5: List of Script File

File Name	Description	Note
_cachestatus.js	checks cache status on the browser	taken from http://www.thecssninja.com/javascript/how-to-create-offline-webapps-on-the-iphone (Ryan., 2009)
CheckConnection.js	checks current network or Internet connectivity	
control.js	chemical room viewing control	
elabel.js	displays marker label on Google map	taken from Google map API library (Williams, 2009) - http://econym.org.uk/gmap/elabel.js
Encryption.js	JavaScript encryption code based on PHP library	taken from http://phpjs.org/functions/sha1 (Zonneveld, 2009)
jquery.ondemand.js	AJAX - JavaScript Library	taken from http://jquery.com/ (Resig, 2009)
jquery-1.3.2.min.js	AJAX - JavaScript Library	taken from http://jquery.com/ (Resig, 2009)
js/Login.js	calls logout if the logout button is clicked	
js/SelectCHWDB.js	queries SQLite database on the client side	
loaddata.js	parse geo-location XML data to Google map	
pagetransition.js	page transisition functions	
popupcontrol.js	chemical viewing pop up control	
SetupDB.js	creates SQLite database on the client side	
String.js	Trims space of user input in the input form	
tabbedmaxcontent.js	displays multiple tabs off a marker on Google map	taken from Google map API library (Liu, 2009)
touchOrient.js	screen orientation	
utils.js	Google map loading functions and some common functions	
Validator.js	user login authentication validation	
JavaScript Geographic locations	longitude and latitude capture Geo-location	Safari Reference Library: Getting Geographic Locations (Apple Inc, 2009)

Table 6: List of Style Sheet, XML, XSL File

File Name	Description	Note
building/iPhone.css	style sheet for iPhone/iPod Touch screen chemical viewing page	CHW Web Server
building/style.css	style sheet for floor map on chemical viewing page	CHW Web Server
chemical.xml	chemical list retrieved from MySQL database through CHW	CHW Web Server
currentlocation.xml	location list retrieved from MySQL database through CWH	CHW Web Server
desktop.css	style sheet desktop screen size web browser	CHW Web Server
form/formDesktop.css	style sheet for desktop screen login page (i.e., login input form)	CHW Web Server
form/formIphone.css	style sheet for iPhone/iPod Touch screen login page (i.e., login input form)	CHW Web Server
iPhone.css	style sheet for iPhone/iPod Touch screen size web browser	CHW Web Server
map/desktop.css	style sheet for desktop screen map page (i.e., resource tracking)	CHW Web Server
map/iPhone.css	style sheet for iPhone/iPod Touch screen map page (i.e., resource tracking)	CHW Web Server
room.xsl	parse XML chemical listing file	

7. References

- Addo, H. (2009, April 8). Ushahidi Was At W3C Workshop In Maputo. In *Ushahidi* [Ushahidi Open-Source Project Blog]. Retrieved September 1, 2009, from Ushahidi.com website: http://blog.ushahidi.com/index.php/2009/04/08/ushahidi-was-at-w3c-workshop-in-maputo/
- Ambler, S. W. (2003). Chapter 12 - Database Refactoring. In *Agile Database Techniques: Effective Strategies for the Agile Software Developer*. Retrieved from http://common.books24x7.com.ezproxy.rit.edu/book/id_11189/book.asp, John Wiley & Sons
- Andrews, M., & Whittaker, J. A. (2006). *How to break Web software: functional and security testing of Web applications and Web services*. Upper Saddle River, NJ: Addison-Wesley.
- Apple Inc. (2009, May 24). HTML 5 Offline Application Cache. In *Safari Reference Library* [Safari Client-Side Storage and Offline Applications Programming Guide]. Retrieved August 1, 2009, from http://developer.apple.com/safari/library/documentation/iPhone/Conceptual/SafariJSDatabaseGuide/OfflineApplicationCache/OfflineApplicationCache.html
- Apple Inc. (2009, June 24). Using the JavaScript Database. In *Safari Client-Side Storage and Offline Applications Programming Guide* [Safari Online Reference Library]. Retrieved June 30, 2009, from Apple Inc. website: http://developer.apple.com/safari/library/documentation/iPhone/Conceptual/SafariJSDatabaseGuide/UsingtheJavascriptDatabase/UsingtheJavascriptDatabase.html#//apple_ref/doc/uid/TP40007256-CH3-SW1

- Apple Inc. (2009, August 4). Getting Geographic Locations. In *Safari Web Content Guide* [Safari Reference LIbrary]. Retrieved August 8, 2009, from Apple Inc. website: http://developer.apple.com/safari/library/documentation/AppleApplications/Reference/SafariWebContent/GettingGeographicalLocations/GettingGeographicalLocations.html
- Azad, K. (2007, April 4). How To Optimize Your Site With GZIP Compression. In *BetterExplained* [Programming & Web Development Explanations]. Retrieved April 17, 2009, from BetterExplained.com website: http://betterexplained.com/articles/how-to-optimize-your-site-with-gzip-compression/
- Ballad, T., & Ballad, W. (2009). *Securing PHP web applications*. Upper Saddle River, NJ: Addison-Wesley.
- Banks, K. (2008, September/October). Mobile Telephony and the Entrepreneur: An African Perspective. *Microfinance Insights*, 25-26. Retrieved September 10, 2008, from http://www.kiwanja.net/miscellaneous/Microfinance-Insights-kiwanja-2008.pdf
- Bellavista, P., & Corradi, A. (2007). Mobile Middleware for Location-Dependent Services. In *The Handbook of Mobile Middleware*. Retrieved November 5, 2008, from http://library.books24x7.com.ezproxy.rit.edu/toc.asp?bookid=16462
- Berka, J. (2008, January 22). PhoneGap tool provides JavaScript access to iPhone features. In *PhoneGap tool provides JavaScript access to iPhone features - Ars Technica* [Internet Article]. Retrieved January 31, 2009, from http://arstechnica.com/apple/news/2008/10/phonegap-tool-provides-javascript-access-to-iphone-features.ars

- B'Far, R. (2005). *Mobile Computing Principles: Designing and Developing Mobile Applications with UML and XML*. Retrieved January 22, 2009, from http://library.books24x7.com.ezproxy.rit.edu/book/id_18281/viewer.asp?bookid=18281&chunkid=0000000001
- Bisson, J., & Saint-Germain, R. (2007). The BS 7799/ISO 17799 Standard for a better approach to information security. *InfoDom / Callio Secura 17799*, white paper. Retrieved April 18, 2008, from Callio Technologies website: http://www.infodom.hr/calliosecura/materijali/White_Paper_ISO_17799_en%5B1%5D.pdf
- Carroll, J., & Broadhead, R. (2001). *Selling online: how to become a successful e-commerce merchant*. Chicago, U.S.A: Dearborn Trade.
- Chen, S., Dunagan, J., Verbowski, C., & Wang, Y.-M. (2005, February). A Black-Box Tracing Technique to Identify Causes of Least-Privilege Incompatibilities. *Network and Distributed System Security Symposium (NDSS)*.
- Dawson, M., Winterbottom, J., & Thomson, M. (2007). *IP Location*. Retrieved November 7, 2008, from http://library.books24x7.com.ezproxy.rit.edu/book/id_18220/viewer.asp?bookid=18220&chunkid=586541598
- Denninger, C. (2009). [User Interview with RIT Public Safety Department]. Unpublished raw data, Rochester Institute of Technology (RIT), Rochester, New York.
- Douangboupha, P., Jett, R., Dangelantonio, D., Sanjanwala, A., & Thomas, L. C. (n.d.). Event Helper. In *Event Helper* [4002.590.03 - Web App Develop Iphone Group Project]. Retrieved April, 2009, from 4002.590.03 - Web App Develop Iphone website: http://www.morethanmachine.com/iphonecourse/

- DynamicDrive. (1998-2008). Online Image Optimizer- GIF, JPG, and PNG. In *DynamicDrive* [Online Image Optimizer- GIF, JPG, and PNG]. Retrieved April 17, 2009, from http://tools.dynamicdrive.com/imageoptimizer/index.php
- Erl, T. (2009). Service Interaction Security Patterns. In *SOA design patterns* (pp. 640-667). Upper Saddle River, NJ: Prentice Hall.
- Ferguson, S. (2008, June 30). Dell Crafts Security Services for Laptops. In *eWeek* [Online Technology Report]. Retrieved November 4, 2008, from Ziff Davis Enterprise Holdings Inc. eWEEK and Spencer F. Katt website: http://www.eweek.com/c/a/Security/Dell-Crafts-Security-Services-for-Laptops/
- Ferraro, A. (1998). Trust: Building It and Keeping It. In *Electronic Commerce: The Issues and Challenges to Creating Trust and a Position Image in Consumer Sales on the World Wide Web*. Retrieved May 9, 2008, from Firstmodnay website: http://www.firstmonday.org/issues/issue3_6/ferraro/index.html
- Frakes, D. (2009, July 15). Inside iPhone 3.0's Remote Wipe feature [Newsgroup message]. Retrieved from http://www.macworld.com/article/141605/2009/07/remotewipe.html
- Funk, J. L. (2004). *Mobile disruption: the technologies and applications driving the mobile Internet*. Hoboken, N.J: Wiley.
- Furht, B., & Ilyas, M. (2003). *Wireless Internet Handbook: Technologies, Standards, and Applications*. Retrieved November 8, 2008, from http://library.books24x7.com.ezproxy.rit.edu/toc.asp?bookid=5970
- Garfinkel, S., & Cranor, L. F. (2005). *Security and usability: designing secure systems that people can use.* Beijing Farnham: O'Reilly.

- Geddis, S., Adler, M., & Brouwer, M. (2009, June). *iPhone Security Best Practices real-world deployment considerations (Session 625)*. Lecture, Session presented at WWDC 2009, SF, U.S.
- Gehani, N. (2002). *XML Technologies*. Retrieved May 5, 2009, from Books24x7 website: http://common.books24x7.com.ezproxy.rit.edu/book/id_12195/book.asp
- Gehani, N. (2007). *The Database Book: Principles & Practice Using MySQL*. Retrieved August 1, 2009, from Books24x7 website: http://common.books24x7.com.ezproxy.rit.edu/book/id_14899/book.asp
- Golding, P. (2008). IP-Centric Mobile Ecosystem and Web 2.0. In *Next generation wireless applications: creating mobile applications in a Web 2.0 and Mobile 2.0 world* (2nd ed., pp. 85-128). Hoboken, NJ: J. Wiley & Sons.
- Golding, P. (2008). J2EE Presentation Layer. In *Next generation wireless applications : creating mobile applications in a Web 2.0 and Mobile 2.0 world* (2nd ed., pp. 207-281). Hoboken, NJ: J. Wiley & Sons.
- Google. (2008). Services - Google Maps API - Google Code. In *Google Maps API* [Google Maps API Reference]. Retrieved November 9, 2008, from Google website: http://code.google.com/apis/maps/documentation/services.html#XML_Requests
- Google. (2009). Geolocation API. In *Geolocation API - Gears API - Google Code* [documentation]. Retrieved January 31, 2009, from Google website: http://code.google.com/apis/gears/api_geolocation.html#getCurrent
- Google. (2009). Google Latitude. In *Google Latitude* [product description]. Retrieved February 17, 2009, from Google website: http://www.google.com/latitude/intro.html

- Gregorio, J. (2004, December 1). *How to create a REST protocol*. Retrieved December 8, 2008, from http://www.xml.com/pub/a/2004/12/01/restful-web.html
- Hillborg, M. (2002). *Wireless XML Developer's Guide*. Retrieved November 8, 2008, from http://library.books24x7.com.ezproxy.rit.edu/book/id_3793/viewer.asp?bookid=3793&chunkid=0000000001
- Hoffman, P. (1999, October). Accommodating Color Blindness. *The Usability SIG Newsletter, 6*(2). Retrieved from http://www.stcsig.org/usability/newsletter/9910-color-blindness.html
- Hu, W. C. (2009). *Internet-enabled handheld devices, computing, and programming : mobile commerce and personal data applications*. Hershey, PA: Information Science Reference.
- IBM. (n.d.). Global Technology Services for Information On Demand Global Technology Services providing a highly-available, reliable, dynamic infrastructure for Information On Demand. In *Information on Demand* [IBM Software Information Management Page]. Retrieved September 1, 2009, from http://www-01.ibm.com/software/data/information-on-demand/global-services.html
- Jefferson, T. (2009, February). *FriendFinder Web.* RIT Information Technology Web Application Lecture presented at 4002.590.03 - Web App Develop Iphone, Rochester Institute of Technology (RIT).

- Jensen, C. S., & Tiesyte, D. (2008, April). Efficient Cost-Based Tracking of Scheduled Vehicle Journeys. *IEEExplore*, 9-16. Retrieved September 12, 2008, from The Ninth International Conference on Mobile Data Management website: http://ieeexplore.ieee.org/iel5/4511414/4511415/04511429.pdf?isnumber=4511415&prod=CNF&arnumber=4511429&arSt=9&ared=16&arAuthor=Tiesyte,+D.;+Jensen,+C.S.
- Johnson, D., White, A., & Charland, A. (2007). *Enterprise AJAX: strategies for building high performance web applications*. Upper Saddle River, NJ: Prentice Hall.
- Kambourakis, G. (2008). On Mobile Wiki Systems Security. In S. Furnell, S. Katsikas, J. Lopez, & A. Patel, *Securing information and communications systems: principles, technologies, and applications Artech House information security and privacy series* (pp. 323-334). Boston: Artech House.
- Katsaros, D., Nanopoulos, A., & Manolopoulos, Y. (2005). Location-Based Services. In *Wireless Information Highways* (section IV - location-based services). Retrieved from http://library.books24x7.com.ezproxy.rit.edu/toc.asp?bookid=9183
- Killeen, J. P., Chan, T. C., Buono, C., Griswold, W. G., & Lenert, L. A. (2006). A Wireless First Responder Handheld Device for Rapid Triage, Patient Assessment and Documentation During Mass Casualty Incidents. *AMIA Annu Symp Proc*, 429–433. Retrieved from http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1839472
- Kim, F., & Skoudis, E. (2009). AppSec - Protecting Your Web Apps: Two Big Mistakes and 12 Practical Tips to Avoid Them. *the SANS Institute Reading Room, 1.1*.
- King, A. B. (2008). *Website optimization*. Farnham: O'Reilly.

- Kinnan, K. (2007, November 20). Virtual Earth API: Adding KML or GeoRSS Layers to the Map [Web log message]. Retrieved from http://blogs.msdn.com/keithkin/archive/2007/11/20/virtual-earth-api-adding-kml-or-georss-layers-to-the-map.aspx
- Kiwanja.net. (n.d.). FrontlineSMS. In *FrontlineSMS* [Open Source Short Messaging Service for Non-Governmental Organizations (NGOs)]. Retrieved September 6, 2009, from http://www.frontlinesms.com
- Kleimola, J. (2008). A RESTful Interface to a Mobile Phone. *Helsinki Institute for Information Technology HIIT*. Retrieved December 7, 2008, from Ubiquitous Interaction Group, Helsinki University of Technology, Finland website: http://www.hiit.fi/~kleimola/net/pamp/pampREST_paper.pdf
- Kung, H.-Y., Ku, H.-H., Wu, C.-I., & Lin, C.-Y. (2008, January). Intelligent and situation-aware pervasive system to support debris-flow disaster prediction and alerting in Taiwan. *Journal of Network and Computer Applications, 31*(1), 1-18 . Retrieved September 12, 2008, from The ACM Portal 2008 website: http://portal.acm.org/citation.cfm?id=1321783.1321931&coll=&dl=ACM
- Laineste, J. (2009). *Terminal-Based Positioning*. http://www.nutiteq.com/: Nutiteq. http://www.slideshare.net/jaakl/terminalbased-mobile-positioning-overview-presentation
- Lehikoinen, J., Aaltonen, A., Huuskonen, P., & Salminen, I. (2007). *Personal content experience managing digital life in the mobile age*. Chichester, England; Hoboken, NJ: John Wiley.
- Liu, N. (2009). *Google code gmaps-utility-library* (Version 1.0) [Data file]. Retrieved April 4, 2009, from Google Code website: http://code.google.com/p/gmaps-utility-library/source/browse/trunk/tabbedmaxcontent/1.0/src/tabbedmaxcontent.js?r=139

- Mallick, M. (2003). *Mobile and Wireless Design Essentials*. Retrieved November 8, 2008, from http://library.books24x7.com.ezproxy.rit.edu/book/id_5394/viewer.asp?bookid=5394&chunkid=0000000001
- Mark, D., & LaMarche, J. (2009). Where Am I? Finding Your Way with Core Location. In *Beginning iPhone Development: Exploring the iPhone SDK* (pp. 429-439). USA: Apress.
- Mehta, N. (2008). *Mobile Web Content*. Birmingham: Mumbai: PACKT.
- Mexens. (2005-2008). How it works. In *Peer-to-peer wireless positioning* [product description]. Retrieved January 31, 2009, from http://www.navizon.com/FullFeatures.asp#NavizonWireless
- Monson, P. (2007). Chapter 1 - Introduction to IBM WebSphere Dashboard Framework. In *Building Composite, Role-Based Dashboards using WebSphere Dashboard Framework. IBM Redbooks* (ibm redbooks). Retrieved from http://common.books24x7.com.ezproxy.rit.edu/book/id_23807/book.asp
- Naresh, V., Pingali, P., Varma, V., Krishna, M., & Venkata, P. (2008, November). *Location Based Web Search on GSM/GPRS Mobile Phones* (Rep. No. IIIT/TR/2008/91). WWW-2006 Developer Track, 23-26 May 2006 to be held in Edinburgh, Scotland, UK: International Institute of Information Technology Hyderabad (Deemed University).
- Nexus. (2007, October 25). Preventing Cross-Site Request Forgery (CSRF)/Session Riding. In *Playhack Security Project* (http://nexus.playhack.net) [Secure Web Apps]. Retrieved August 1, 2009, from PLAYHACK.net website: http://files.playhack.net/papers/preventcsrf.txt

- Nurminen, J. K., Wikman, J., Kokkinen, H., Muilu, P., & Grönholm, M. (2008, January). Drupal Content Management System on Mobile Phone. *IEEE Xplore*, 1228-1229.
- In Proceeding of the 5th IEEE CCNC (Consumer Communications and Networking Conference) 2008 proceedings, Las Vegas, NY, USA
- Nützel, J., & Beyer, A. (2006). Towards Trust in Digital Rights Management Systems. *Lecture Notes in Computer Science, Volume 4083/2006*, 162-171.
- Olla, P. (2008). Global Navigation and Satellite Systems and Services. In *Commerce in Space: Infrastructures, Technologies, and Applications* (chapter v). Retrieved November 8, 2008, from http://library.books24x7.com.ezproxy.rit.edu/book/id_23221/viewer.asp?bookid=23221&chunkid=721877683
- O'Neill, M., Hallam-Baker, P., Cann, S. M., Shema, M., Simon, E., Watters, P. A., & White, A. (2003). *Web Services Security*. Retrieved December 13, 2008, from http://library.books24x7.com.ezproxy.rit.edu/
- Pashtan, A. (2005). *Mobile Web Services*. Cambridge, UK; New York: Cambridge University Press.
- PennState. (2008, March 18). Human Factors and HCI [Web log message]. Retrieved from Human Factors and HCI: http://www.personal.psu.edu/cwc5/blogs/coursedesign/
- Pernul, G. (2008). Data-Centric Applications. In S. Furnell, S. Katsikas, J. Lopez, & A. Patel, *Securing information and communications systems: principles, technologies, and applications Artech House information security and privacy series* (pp. 87-103). Boston: Artech House.

- Priebe, T. (2008). Authorization and Access Control. In S. Furnell, S. Katsikas, J. Lopez, & A. Patel, *Securing information and communications systems: principles, technologies, and applications Artech House information security and privacy series* (pp. 61-85). Boston: Artech House.
- Pritchard, E. (2003). WebRSA. In *WebRSA* [Implementation of PKCS #1 v2.1 RSA CRYPTOGRAPHY STANDARD (RSA Laboratories, June 14, 2002)]. Retrieved August 1, 2009, from Guardian Unlimited website: http://webrsa.sourceforge.net/
- Purvis, M., Sambells, J., & Turner, C. (2006). *Beginning Google Maps Applications with PHP and Ajax: From Novice to Professional*. Retrieved November 7, 2008, from http://library.books24x7.com.ezproxy.rit.edu/book/id_25551/viewer.asp?bookid=25551&chunkid=0000000001
- Renaud, K. (2005). Evaluating Authentication Mechanisms. In S. Garfinkel & L. F. Cranor, *Security and usability: designing secure systems that people can use* (pp. 103-128). Beijing Farnham: O'Reilly.
- Resig, J. (2009, May 19). jQuery JavaScript Library v1.3.2. In *jQuery JavaScript Library v1.3.2* [jQuery JavaScript Library v1.3.2]. Retrieved May, 2009, from http://jquery.com/
- Richards, R. (2006). *Pro PHP XML and Web Services*. Retrieved December 8, 2008, from http://library.books24x7.com.ezproxy.rit.edu/book/id_14639/viewer.asp?bookid=14639&chunkid=0000000001
- RIT. (2008, June). Floor plans. In *RIT Campus Map* [Floor plan map in PDF version]. Retrieved February, 2009, from Rochester Institute of Technology (RIT) website: http://www.ntid.rit.edu/cat/summit/pdf/FloorMap.pdf

- Ritter, A. (2009, June 13). A better include function for JavaScript [Web log message]. Retrieved from http://www.bytestrom.eu/blog/2009/0605a_better_include_function_for_javascript
- Ryan. (2009, April 28). How to create offline webapps on the iPhone. In *The CSS Ninja* [CSS, Javascript & xhtm Posts]. Retrieved July, 2009, from thecssninja.com website: http://www.thecssninja.com/javascript/how-to-create-offline-webapps-on-the-iphone
- Sauter, M. (2009). Mobile Web 2.0, Applications and Owners. In *Beyond 3G: bringing networks, terminals and the Web together: LTE, WiMAX, IMS, 4G devices and the mobile Web 2.0* (pp. 273-343). Chichester, West Sussex, U.K. : John Wiley .
- Silva, C. D. (2006, December). Sahana Free and Open Source Disaster Management Project. *Sahana Free and Open Source Disaster Management System Project Overview DRAFT 0.9, DRAFT 0.9*.
- Singh, M. P., & Huhns, M. N. (2005). *Service-Oriented Computing: Semantics, Processes, Agents*. Retrieved December 8, 2008, from http://library.books24x7.com.ezproxy.rit.edu/book/id_11288/viewer.asp?bookid=11288&chunkid=0000000001
- SiteReportCard.com. (2002-2007). Image Optimization. In *SiteReportCard* [A free utility online for Image Optimization]. Retrieved April 17, 2009, from http://sitereportcard.com/imagereducer.php
- Steiner, T. (Writer). (2008). REST Describe & Compile, or WADL2Anything [Motion picture]. Google Tech Talks. Retrieved December 12, 2008, from Google Tech Talks website: http://www.youtube.com/watch?v=hZ2EtAEBpq0

- Thus, T. (2008, April 23). *Inbox Innovation, Web Optimization by BlueTie*. Lecture presented at Rochester Institute of Technology (RIT), Rochester Institute of Technology (RIT).
- *Ushahidi* [An opensource project for handheld device where people can submit crisis information to the website.]. (n.d.). Retrieved September 11, 2008, from Ushahidi.com website: http://www.ushahidi.com/
- Vance, A. (2009, June 13). The iPhone 3GS and Forensics: Encryption Changes the Game? [Web log message]. Retrieved from http://anthonyvance.com/blog/forensics/iphone_encryption/
- Verclas, K. (2008, September 26). *ITU Predicts 4 Billion Mobile Subscriptions at the End of 2008* [Blog]. Retrieved November 8, 2008, from http://mobileactive.org/itu-predicts-4-billion-mobile-subscriptions-end-2008
- VeriSign. (2005). Establish Trust to Protect and Grow Your Online Business Preview. In *VeriSign* [In VeriSign SSL Guides]. Retrieved May 8, 2008, from http://www.verisign.com/Resources/SSL_Services_Guides/page_002771.html
- Warsi, A. (2007). 7 usability guidelines for websites on mobile devices. In *Webcredible User Experience Research and Design* [Website User Usability]. Retrieved November 8, 2008, from http://www.webcredible.co.uk/user-friendly-resources/web-usability/mobile-guidelines.shtml
- Williams, M. (n.d.). *Google Maps API Tutorial* (Version 1.8) [Data file]. Retrieved April 4, 2009, from http://econym.org.uk/gmap/elabel.js

- Wu, S.-L., & Tseng, Y.-C. (2007). *Wireless Ad Hoc Networking—Personal-Area, Local-Area, and the Sensory-Area Networks* (S.-L. Wu & Y.-C. Tseng, Eds.). Retrieved January 30, 2009, from http://library.books24x7.com.ezproxy.rit.edu/book/id_26492/viewer.asp?bookid=26492&chunkid=0000000001
- Zhang, Q., Mayes, K., & Markantonakis, K. (2005, November). A user-centric m-payment solution. *IEEE Mobility Conference 2005*. Retrieved September 10, 2008, from IEEE Xplore website: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1656674
- Zonneveld, K. V. (2009). PHP.JS. in *PHP.JS*. Retrieved July 30, 2009, from dual licensed under the MIT and GPL (GPL-LICENSE.txt) licenses website: http://phpjs.org/functions/